Formal Verification Techniques using Quantum Process Calculus

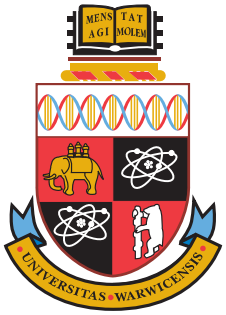
Timothy A. S. Davidson

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

January 2012

Supervisor: Dr. Rajagopal Nagarajan



Department of Computer Science
University of Warwick
Coventry CV4 7AL
United Kingdom

Contents

List of Figures

Acknowledgements

First, I would like to thank my supervisor, Rajagopal Nagarajan, for the opportunity to study a very interesting and frequently perplexing topic. His guidance and encouragement has been invaluable.

I am immensely grateful to Simon Gay, who has provided support and guidance from afar. My trips to Glasgow have always sparked renewed enthusiasm and inspiration, and Simon's input has been crucial in writing this thesis.

Thanks also go to Nick Papanikolaou and Hynek Mlnřík with whom I have had the pleasure of working alongside. Our many discussions, both technical and non-technical, have provided encouragement as well as entertainment. From them I have learned many things which have helped move my research along, and I am very grateful for their help.

I have enjoyed being involved with Warwick Student Cinema, and I appreciate all the hard work of everyone at the society for making it such an exciting and rewarding distraction from the depths of my thesis. In particular, I would like to thank Rachel, Martin, Nick, Matt and Amanda for their support, encouragement, and the many pub trips.

I wish to thank Brian for his many words of wisdom, and my parents for their love and support, and because it is always reassuring to know they are thinking of me.

And finally, to Emma, who has stood by my side every step of the way, thank you for everything.

Declaration

The work described in this thesis is original work and is based on collaborations with Rajagopal Nagarajan, Simon Gay, Hynek Mlnařík and Nikolaos Papanikolaou. Some of the material presented in Chapter 3 was presented in the following poster:

- T. Davidson, S. J. Gay and R. Nagarajan. Verifying Quantum Teleportation with Quantum Process Calculus. *Fifth Conference on the Theory of Quantum Computation, Communication and Cryptography.* 2010.

The material presented in Chapter 6 has been published in

- T. Davidson, S. J. Gay, H. Mlnařík, R. Nagarajan and N. Papanikolaou. Model Checking for Communicating Quantum Processes. *International Journal of Unconventional Computing* 8(1):73-98, 2012.

Abstract

Quantum communication is a rapidly growing area of research and development. While the successful construction of a large-scale quantum computer may be some years away, there are already commercial implementations of secure communication using quantum cryptography. The application of formal methods to classical communication and cryptographic systems has been very successful, and is now widely used in industry by organisations such as Intel, Microsoft and NASA. There is reason to believe that similar benefits can be expected for the verification of quantum systems.

In this thesis, we focus on the use of process calculus, specifically Communicating Quantum Processes (CQP), for the analysis of quantum protocols. Congruence relations are an important aspect of process calculus, since they provide the foundation for equational reasoning. Previous work on congruence relations for quantum processes excluded the classical information arising from measurements, and was therefore unable to analyse many of the interesting known quantum communication protocols. Developing a congruence relation for general quantum processes is difficult because of the interaction between measurement, entanglement and parallel composition.

We define a labelled transition relation for CQP in order to describe external interactions. Based on this semantics, we define a notion of observational equivalence for CQP processes, namely *probabilistic branching bisimilarity*. We find that this relation is not preserved by parallel composition, however we are able to gain a deeper understanding of the link between probabilistic branching and measurement. Based on this newfound understanding, we present a novel semantics for quantum processes, combining mixed quantum states with probabilistic branching. With respect to this new semantic model, we define *full probabilistic branching bisimilarity* and prove that it is a congruence. We use this congruence relation to discuss an axiomatic approach to the verification of quantum processes. The quantum teleportation protocol is used as a primary example throughout, and we prove that it is congruent to a quantum channel.

We define a translation from CQP to the Quantum Model Checker (QMC) in order to provide automated verification techniques using CQP specifications. We prove that this translation preserves the semantics of CQP processes, thereby enabling a multi-faceted approach to formal verification by enhancing the manual techniques of process calculus with the benefits of model checking.

Like mathematics, computer science will be somewhat different from the other sciences, in that it deals with artificial laws that can be proved, instead of natural laws that are never known with certainty.

— Donald Knuth

1
Introduction

The rapidly growing area of quantum information science has the potential to transform computing in an unprecedented manner, offering new paradigms, increased computational power, and absolute security guarantees. Arguably one of the most significant promises offered by quantum phenomena is the potential for unconditionally secure communication.

Cryptography has been used for thousands of years to allow users to keep their communications private. The field has gained significant momentum in recent decades, particularly as computational techniques have been applied both for cryptography and for cryptanalysis. Despite many advances, key distribution, the process by which two or more users agree on a shared secret (the *key*), remains a fundamental problem in cryptography. Often, the ability to generate a shared secret (which could be through a private meeting or a trusted third party) obviates the need to use cryptography since a secure channel is already available. The development of *public key cryptography* has provided a solution to the key distribution problem, enabling many of the secure communications made in the present day. Internet communications are one obvious and major use of public key cryptography, however the solution is not unconditionally secure.

Public key cryptography is based on *trapdoor functions* [Diffie and Hellman 1976]; these so called mathematical functions have the characteristic that they can be easily computed one way, however without extra information (called the *trapdoor*), the reverse computation is significantly harder. Prime factorisation and discrete logarithms have both been used as the basis for trapdoor functions, leading to the well–known RSA [Rivest et al. 1978], Rabin [Rabin 1979] and ElGamal [ElGamal 1985] public key cryptosystems. The fact remains, that although computationally hard, given sufficient resources these trapdoor functions can be reversed. The increase in computational power over time and the increasing ability for distributed computing are

significant concerns for public key cryptography, particularly when it is used to store secret information over a period of time.

Ironically, quantum computing poses a significant threat to many of the public key cryptosystems in use today. The widespread interest in quantum computation was fueled by Shor's discovery in 1994 of an efficient quantum algorithm for prime factorisation [Shor 1994]. Such an algorithm has the potential to obliterate the effect of public key cryptosystems, stopped only by the lack of technology. Practical quantum computers are still a long way from becoming reality, due to the difficulties of implementing registers and memory for quantum information. Laboratory setups have to date implemented Shor's algorithm to factor the number 15.

On the other hand, the promise of secure communication has already been delivered by quantum cryptography; key distribution networks have been tested in Vienna [Poppe et al. 2004] and Boston [Elliott et al. 2005], and quantum cryptographic systems are commercially available from several companies (e.g. MagiQ Technologies, ID Quantique, Toshiba, NEC). *Quantum key distribution* is a solution to the key distribution problem that enables two parties to generate a shared secret key, whilst remaining confident that a third party has no information about their key. To detect the presence of an eavesdropper, Bennett and Brassard's BB84 key distribution protocol [Bennett and Brassard 1984] uses the property of quantum mechanics that an observation of the quantum system irreversibly disturbs the state. Even if an eavesdropper was able to obtain some information during the initial exchange, the two parties can use secret key reconciliation [Brassard and Salvail 1994] and privacy amplification [Bennett et al. 1995] to invalidate that information.

A similar key distribution protocol due to Ekert [1991] uses quantum entanglement, in particular so-called *Einstein-Podolsky-Rosen* (EPR) [Einstein et al. 1935] pairs of qubits, to generate a shared key. Statistical analysis based on Bell's theorem [Bell 1964] allows the two parties to detect the presence of an eavesdropper. Bennett et al. [1992b] later showed that Bell's theorem is not a requirement for key distribution using entanglement.

Key distribution is not the only application of quantum communication, although it is one of the most interesting and relevant. Several other interesting protocols that fall under the remit of quantum cryptography have also been developed, including quantum bit commitment [Kent 2003], coin-flipping [Bennett and Brassard 1984], and quantum money [Wiesner 1983].

1.1 Context

1.1.1 Quantum Information

The difference between classical and quantum information is determined by the physical systems with which they are associated. Around the 1920's, the theory of *quantum mechanics* was founded, out of the need to explain the properties of atomic particles and light. At these small scales (below around 10^{-7}m), far smaller than is visible to the naked eye, existing theories of physics (now known as *classical physics*) are unable to explain the observations.

Quantum information theory is concerned with the properties of atomic particles and how they can be used for the representation and storage of data. At first, the abnormal behaviour of atomic particles seems to present a significant challenge for data processing at these minute scales. However the potential of using such phenomena instead of viewing them as a barrier was realised by Wiesner [1983], who proposed the representation of data using polarised photons.

Atoms, photons and electrons are examples of *quantum systems*, which can be attributed with physical properties such as position, momentum, spin, and polarisation. These properties describe the *state* of the system, and provide the means to represent information. Analogous to the way in which the state of a switch (*off* or *on*) represents a classical bit (0 or 1), we can associate a quantum value (let us write these as $|0\rangle$ and $|1\rangle$) to orthogonal polarisations of a photon (e.g. 0° and 90°).

Quantum information differs from classical information as a result of the, often counter-intuitive, behaviour of the underlying quantum systems. The distinctive properties of quantum systems include superposition, entanglement, non-locality, probabilistic measurement and non-cloneability. The *superposition principle* states that, if a system can be in one of several given states, then it can also be in a simultaneous combination of these states. This enables so-called *quantum parallelism*, in which computations can be applied to many states at the same time. *Entanglement* is a property of quantum systems that is used in many communication protocols. It describes the possibility that two or more quantum systems may be linked in such a way that it is impossible to fully describe the systems individually. *Non-locality* describes the property that entanglement is preserved between quantum systems even when they are physically separated.

The state of a quantum system is discovered through *measurement*, however measurement produces a probabilistic outcome based on the true state of the system. Furthermore, the act of measurement causes the system to collapse into the observed state, and is therefore considered a *destructive* operation. The *no-cloning theorem* [Dieks 1982; Wootters and Zurek 1982] arises from the probabilistic and destructive nature of measurement, and states that an unknown quantum state cannot be

duplicated. Although this may be considered a hindrance to computation, the non-cloneability property plays an essential role in many cryptographic protocols.

Quantum information is particularly fragile, since any interaction with the environment can result in the state collapsing. This phenomena is called *decoherence*, and must be prevented in order to carry out computation and communication. Decoherence is one of the main obstacles in building large-scale quantum computers, however developments in quantum error correction can help to mitigate this problem.

1.1.2 Quantum Computing

The idea of using quantum physics for computation was first alluded to in 1982 by Feynman [1982], who described the difficulties of simulating quantum mechanical systems on classical computers. In 1985, Deutsch [1985] attempted to define a computational device that is capable of simulating an arbitrary physical system. This device, called the *universal quantum computer*, is meant to be the quantum equivalent of the model on which classical computer science is based; the *universal Turing machine*.

Although of little practical use, the Deutsch-Jozsa algorithm [Deutsch and Jozsa 1992] provided one of the first examples that quantum algorithms could offer an exponential speedup over classical computation. Inspiration from this work lead to the practical algorithms developed by Shor [1994] and Grover [1996].

The generation of widespread interest in quantum computing is often attributed to Shor's algorithms for solving prime factoring and discrete logarithm problems. These algorithms have many practical applications, and are particularly significant with respect to the security of public key cryptosystems. A third class of quantum algorithms, aside from simulation and *quantum Fourier transform*-based algorithms such as Shor's, are unstructured search algorithms. Grover's algorithm [Grover 1996] is the first example of an algorithm in this class, enabling the efficient search of unstructured data.

Realising these benefits of quantum computation is still several years down the line. Physicists still have to develop the technology for the storage and accurate manipulation of quantum systems on a large scale, without causing decoherence.

1.1.3 Quantum Communication

In contrast to quantum computation, the ability to use quantum mechanics for communication is already a practical technology. That is not to say it is ready for widespread applications; indeed, the development of practical communication networks require implementations of *quantum memory* and *entanglement swapping*. However, there are commercial devices available from companies including idQuan-

tique, MagiQ, Toshiba and NEC, which are designed for point-to-point secure communication using fibre-optic cable.

The idea of quantum cryptography was first put forward by Wiesner in the late 1960's, although this was not published until 1983 [Wiesner 1983]. Meanwhile, Bennett and Brassard [1984] developed this idea, resulting in the BB84 *Quantum Key Distribution (QKD)* protocol. This protocol uses polarised photons to encode bits using either the rectilinear basis, or the diagonal basis which is offset by 45°. Based on the destructive nature of measurement, an eavesdropper will cause random errors if she were to measure in the wrong basis, which can later be detected by the communicating parties. This was later implemented in a laboratory over a distance of less than 1m [Bennett et al. 1992a], however a side-channel in the form of a loud buzzing from the power supply whenever a '1' was sent made the security of QKD irrelevant.

Other protocols for QKD were later developed, including the EPR protocol by Ekert [1991] and the B92 protocol by Bennett et al. [1992b]. The EPR protocol uses entangled qubit pairs, which could come from a third party source, and Bell's theorem [Bell 1964] to detect the presence of an eavesdropper. B92 provides a similar solution, but shows that Bell's theorem is not required.

Although QKD is the major application in quantum cryptography, there are a number of other cryptographic protocols that have been developed. These include *bit commitment* (for example, [Brassard and Crépeau 1991; Kent 2003]), coin flipping [Bennett and Brassard 1984; Berlín et al. 2008], and secret sharing [Markham and Sanders 2008]. Also important are *privacy amplification* [Bennett et al. 1988, 1995] and *secret key reconciliation* [Brassard and Salvail 1994], which can be used respectively, to distill a secret key when an eavesdropper has gained some information, and to establish a key over a noisy quantum channel.

Aside from cryptography, there are several other applications of quantum communication. The most interesting, particularly from an information theoretic point of view, are *quantum teleportation* and *superdense coding* [Bennett et al. 1993; Brassard et al. 1998]. Quantum teleportation is a protocol that enables a quantum state to be transmitted using entanglement and the communication of two classical bits. On the other hand, superdense coding reverses this situation, utilising one qubit to transmit two classical bits of information. These protocols are particularly interesting because it is only possible to store a single bit of information per qubit.

1.2 Motivation

Having introduced quantum information and communication, we now describe the motivation for the development and application of formal methods to quantum communication protocols.

A protocol defines the language, semantics and procedures that enable agents in concurrent and distributed systems to establish, maintain and complete communication [Holzmann 1982]. A well designed protocol aims to allow efficient communication in normal circumstances, and should also permit and respond appropriately to occasional errors, such as packet loss.

A well designed protocol should also be able to recover from more serious situations such as machine failures and malicious attacks. This is particularly relevant for safety- and security-critical protocols, where failures can have serious consequences. Designing protocols that are robust in these situations is a difficult task, and requires a complete exploration of all possible conditions, a feat that may be near impossible for complex protocols.

Formal methods encompass a range of techniques, both manual and automated, for the specification and verification of systems. These techniques are characterised by formal languages and precisely defined semantics that offer systematic and generalised approaches to system analysis. Such tried, tested, and well understood methods can be applied to precisely describe protocols and ensure their design criteria are satisfied in all eventualities.

A prominent example illustrating the use of formal methods is the analysis of the Needham-Schroeder public key authentication protocol [Needham and Schroeder 1978] by Lowe [1996]. Using the process algebra CSP [Hoare 1985] and the automated Failures Divergences Refinement (FDR) model checking tool [Roscoe 1994], Lowe was able to discover a possible attack on the protocol, and subsequently verify a corrected version. This vulnerability had gone undiscovered for 17 years.

The large number of quantum cryptographic protocols makes the application of formal methods to quantum systems particularly important. Indeed, many practical implementations are likely to consist of a selection of basic communication and cryptographic protocols working together. The verification of such systems is facilitated by using *compositional analysis*, in which the individual components can be analysed in isolation.

Quantum cryptographic protocols are designed so that their security relies on fundamental features of quantum theory, such as Heisenberg's uncertainty principle and the non-cloneability of unknown quantum states [Wootters and Zurek 1982], and therefore cannot be compromised even by a quantum computer. Naturally it is necessary to prove that a given protocol is actually secure in this sense. The key result in this area is Mayers' proof [Mayers 2001] of *unconditional security* of the BB84 quantum key distribution protocol. This result, and others of its kind, are extremely significant; however, a mathematical proof of security of a *protocol* does not in itself guarantee the security of an implemented *system*, which typically contains classical components as well. Therefore, it is useful to further analyse such protocols using

alternative methods.

Computer scientists have developed a range of techniques and tools for the analysis and verification of communication systems and protocols; Ryan et al. [2001] survey their application to security. These formal methods have had a major impact and are used industrially by organisations such as Intel, Microsoft, and NASA. It is reasonable to expect similar benefits from applying formal methods to quantum systems.

Communicating Quantum Processes (CQP) [Gay and Nagarajan 2005, 2006] is a *process calculus* based on the π-calculus [Milner et al. 1992] and includes operations for quantum information processing. Initial work on CQP focused on developing a flexible and sound operational semantics and type system, centering around the idea that each quantum bit is a physical resource and therefore owned by a unique process. In this thesis, we develop a theory of behavioural equivalence for CQP, in order to support process-oriented specifications of quantum systems. A major result is that full probabilistic branching bisimilarity [Trčka and Georgievska 2008], defined here for CQP, is a congruence. Congruence has been missing from previous work on quantum process calculus [Feng et al. 2006; Lalire 2006; Ying et al. 2007, 2009], but is important in allowing equational reasoning about process equivalence.

1.3 Contribution

The aim in this thesis is to develop techniques for the formal analysis of quantum protocols, through the adaptation of classical formal methods. In particular, we focus on enhancing quantum process calculus and extensions for automated techniques.

Chapters 3 and 4 address an open problem in the field of quantum process calculus; finding congruence relations for general quantum processes. This involves an analysis of existing relations and consideration of the interplay between quantum information and observational equivalence. An important aspect is the consideration of the physical reality of these algebraic models.

The quantum process calculus CQP is used as the vessel for this work, due to its strong foundation. The quantum teleportation protocol plays a central role, providing a practical application that can be used to analyse the accuracy and potential of our model.

We convert the reduction semantics of CQP to a labelled transition semantics in order to describe external interactions, and based on this we develop a notion of process equivalence, namely *probabilistic branching bisimilarity*. This relation is applied to quantum teleportation, and we prove that the protocol is equivalent to a direct quantum channel, which is in agreement with similar results in other formalisms, e.g. [Abramsky and Coecke 2004; Danos et al. 2007a]. We consider the preservation properties of this bisimilarity with respect to the process constructs of CQP, and find that

it is preserved by all constructs except parallel composition. However, for a small class of processes that includes teleportation, equivalence is also preserved by parallel composition.

We discuss the physical accuracy of the semantics and bisimilarity, and argue that the implementation of probabilistic branching does not respect the observational properties of quantum information. This also suggests that the approaches by Feng et al. [2006]; Lalire [2006]; Ying et al. [2007, 2009] may also be flawed in a similar manner.

In order to address this proposed flaw, we present a novel approach for the operational semantics of quantum processes, that combines probabilistic branching with mixed quantum states. We argue that this provides an accurate model of quantum processes with respect to the laws of quantum mechanics. Furthermore, we prove that full probabilistic branching bisimilarity is preserved by all process constructs with respect to this new semantics, and is therefore a congruence.

In Chapter 5 we discuss an axiomatic approach to verification using this newfound congruence relation. A collection of equational laws which are based upon the properties of quantum information are defined and proved. These laws are applied to quantum teleportation, providing a comparatively simpler proof of equivalence than the long-handed approach in Chapter 4, and thus illustrating the significance of an equational theory.

Chapter 6 develops the framework for a multi-faceted approach to the formal analysis of quantum protocols. We present a translation from CQP to the recently developed quantum model checking tool, QMC [Papanikolaou 2009]. We prove that this translation preserves the semantics of CQP processes, enabling the benefits of both formal techniques to be realised from a single specification.

1.4 State of the Art

In recent years there has been significant development in formal languages and semantic techniques for quantum systems. There are surveys by Selinger [2004b], Gay [2006] and Rüdiger [2007], the latter of which focusses on quantum programming languages. The most relevant developments, with respect to this thesis, are in the field of quantum process calculus, reviewed in Section 1.4.1. We also review developments that focus on semantic analysis and verification of quantum systems.

1.4.1 Quantum Process Calculus

Quantum versions of process calculus, which are designed for describing the interactions between different components of a system, have been developed to complement

other quantum languages. The quantum process calculi that have been developed to date are called QPAlg, CQP and qCCS.

QPAlg (named for Quantum Process Algebra) [Jorrand and Lalire 2004; Lalire and Jorrand 2004] is a language similar to the classical process calculi *Calculus of Communicating Systems (CCS)* [Milner 1989] and Lotos [Bolognesi and Brinksma 1987], with extensions for modelling quantum processes. These extensions include primitives for applying unitary operators, measurements and the ability to send and receive qubits. The quantum state was originally represented as a state vector by Jorrand and Lalire [2004], however the more abstract density matrix representation was used in a later version of QPAlg [Lalire and Jorrand 2004] to allow the description of parts of an entangled qubit register. An operational semantics is given for QPAlg in which labelled transitions are complemented by probabilistic transitions, the latter resulting from quantum measurements. The no–cloning principle is satisfied by conditions on the input and output rules, specifying respectively that the qubit in question should not already be initialised or should be removed from the register. However, it is still possible for multiple processes to use the same qubit internally, which is in contrast to the assumption made in other quantum process calculi that processes correspond to physical systems and qubits are physical resources.

Lalire [2005, 2006] investigates equivalences on processes, in particular defining a *probabilistic branching bisimilarity* based on the branching bisimilarity of van Glabbeek and Weijland [1996] and the probabilistic equivalences of Fokkink [2007] and Andova [1999]. The equivalence is extended from a bisimulation on process states to an equivalence relation on processes and is shown to be preserved by all operators except parallel composition. Two problems that prevent preservation by parallel composition (and hence a congruence) are identified; the restriction of quantum variables to individual processes and the comparison between probabilistic and non-deterministic actions.

Communicating Quantum Processes (CQP) was developed by Gay and Nagarajan [2005] around the same time as QPAlg. The development of CQP followed attempts to use the classical process calculus CCS in combination with the Concurrency Workbench of the New Century (CWB-NC) [Cleaveland and Sims 2009] for the verification of the BB84 quantum key distribution protocol [Nagarajan and Gay 2002]. As part of the same research programme, the classical model checking tool PRISM has also been used for the analysis of quantum systems [Gay et al. 2005; Papanikolaou 2004].

CQP is based on the π-calculus [Milner 1999; Milner et al. 1992] with primitives for quantum information inspired by Selinger's QPL [Selinger 2004a]. The operational semantics of CQP are defined using *reductions* under the assumption that quantum systems are closed to any environment; as such the transmission of qubits is internal and no external communication is considered. Quantum measurements are modelled

with probabilistic transitions, following a similar approach to QPAlg. The most distinctive feature of CQP is the inclusion of a static type system, the purpose of which is to classify classical and quantum data and also to enforce the no-cloning property of quantum information. A full treatment of the type system with associated proofs of soundness and a type checking algorithm is presented by Gay and Nagarajan [2006]. The language has been presented as a solid framework with the ability to easily add new functionality as required, although no process equivalences have been investigated.

The language qCCS, a quantum extension of the classical value-passing CCS, was first proposed by Feng et al. [2007]. One aim of this language is to address the short-comings of QPAlg and CQP with respect to the input and output of quantum states, in particular where entanglement is involved. The language uses probabilistic transitions to deal with measurement, however it doesn't treat these as branching transitions, instead maintaining a distribution over each outcome. No-cloning and similar quantum properties are satisfied by conditions at the syntactic level, leading to a syntax that is more complicated than QPAlg and CQP but does not require a type system. Process equivalences are investigated, namely strong and weak probabilistic bisimilarity, which are shown to be preserved by various operators. The most interesting result is that their equivalences are preserved by parallel composition with processes that do not change the quantum context.

A later version of qCCS [Ying et al. 2007, 2009] excludes classical information in an attempt to better understand quantum processes. Quantum operations are modelled using *superoperators* allowing for the operational semantics to be defined by a non-probabilistic transition system. Several notions of equivalence are considered by Ying et al. [2007, 2009], in particular strong reduction-bisimilarity and approximate (reduction-)bisimilarity. The approximate equivalences are motivated by the potential inaccuracies that may occur in the implementation of quantum gates. Significantly, approximate bisimilarity and approximate reduction-bisimilarity are shown to be preserved by parallel composition in this purely quantum setting. However this result is not sufficient for the analysis of most interesting quantum protocols, many of which involve the interaction of quantum and classical information.

In recent work, Feng et al. [2011] combine principles from both [Feng et al. 2006] and [Ying et al. 2009] resulting in an updated version of qCCS. This latest version models general quantum processes, including the ability to model classical information and maintaining the use of superoperators from [Ying et al. 2009]. Significantly, a weak bisimulation is defined and subsequently proved to be a congruence. An axiomatisation is presented which is based on the adaptation of classical CCS laws, however completeness is not considered. The quantum teleportation and superdense coding protocols are used to illustrate the language and congruence. The results of

Feng et al. [2011] are similar, although independent to the results in this thesis; this is discussed further in Chapter 4.

1.4.2 Verification of Quantum Systems

There have been several, primarily mathematical, proofs of correctness and security of various quantum protocols. Mayer's proof of the unconditional security of quantum key distribution is probably the most prominent of these [Mayers 2001]. There are also negative results, for example, proofs that show unconditionally secure quantum bit commitment is impossible [Lo and Chau 1997; Mayers 1997].

Automated model checking techniques have also been applied to a number of quantum protocols. These include attempts to use existing probabilistic model checkers such as PRISM [Kwiatkowska et al. 2001]. Gay et al. [2005] use PRISM for the analysis of quantum teleportation, superdense coding and an error correction protocol. PRISM is also used by Elboukhari et al. [2010] for the verification of the B92 quantum key distribution protocol [Bennett et al. 1992b]. Recently, the Quantum Model Checker (QMC), a model checking tool designed specifically for quantum systems, has been developed by Papanikolaou [2009]. This tool is restricted to *stabilizer circuits*, which offer efficient simulation on classical computers at the expense of falling short of universal quantum computation. Baltazar et al. [2008] define Quantum Computation Tree Logic (QCTL) and a corresponding model checking algorithm for reasoning about quantum protocols.

1.4.3 Semantics for Analysis of Quantum Systems

Abramsky and Coecke [2004] have developed a category theoretic formulation of the axioms of quantum mechanics. This representation enables the mathematical analysis of *information flow* within quantum systems. For example, they have shown the correctness of teleportation.

Perdrix [2007, 2008] introduces an approach to entanglement analysis using an abstract interpretation framework. This focuses on the evolution of entanglement. Prost and Zerrari [2008] consider a logic based approach to entanglement analysis for functional languages, although only pure quantum states are considered.

Altenkirch and Grattage [2005] have developed a functional quantum programming language QML. A sound and complete equational theory for the measurement-free QML is presented by Altenkirch et al. [2007].

A *measurement calculus* is defined for distributed measurement based quantum computation by Danos et al. [2007b]. In this formalism, a notion of operational equivalence is considered and used to show that quantum teleportation is bisimilar to a direct quantum channel [Danos et al. 2007a].

A process calculus designed for the analysis of quantum security protocols has been developed by Adão and Mateus [2007]. The language implements a cost model and is based on the *quantum random access machine (QRAM)* computational model. They define notions of observational equivalence and computational indistinguishability.

1.5 Outline

This thesis is organised as follows. Chapter 2 presents a review of the relevant background material and concepts. Chapter 3 describes a first attempt at defining a process equivalence for CQP, inspired by the results of QPAlg and qCCS. We define a labelled transition relation for CQP, in order to describe external interactions, and use this to define probabilistic branching bisimilarity for CQP processes. We prove that probabilistic branching bisimilarity is preserved by all constructs except parallel composition. We also prove that quantum teleportation is bisimilar to a direct quantum channel in all contexts. A discussion leads to the observation that these semantics do not respect the laws of quantum mechanics.

Chapter 4 introduces the novel approach of combining mixed states with probabilistic branching; a notion that we call *mixed configurations*. The operational semantics of CQP is redefined using these mixed configurations, and we prove that typing is preserved by the new transition relations. We prove that full probabilistic branching bisimilarity is a congruence with respect to this new semantics.

Chapter 5 presents an equational theory for full probabilistic branching bisimilarity. The equational laws are centered around the observational properties of quantum information, and are illustrated through an application to quantum teleportation. We prove the soundness of this equational theory, and we discuss the issues involved in adapting the expansion lemma from the π-calculus for this relation.

Chapter 6 describes an approach to quantum protocol analysis that combines process calculus and model checking. We define a translation from CQP to the quantum model checking tool QMC, and prove that the semantics of CQP processes is preserved.

We conclude in Chapter 7 with a final review, as well as outlining directions for future work.

2
Background

In this chapter, we review the relevant background concepts, covering quantum mechanics, quantum protocols and process calculus.

2.1 Quantum Mechanics

The theory of quantum mechanics is described by a mathematical formulation that is very different from classical mechanics. In this section, the concepts necessary to describe quantum mechanics are introduced. For further reading, there are many textbooks on the subject, in particular, [Nielsen and Chuang 2000] and [Gruska 1999] are standard references. An account aimed at computer scientists is given by Rieffel and Polak [2000].

2.1.1 Hilbert Spaces

The convention in quantum mechanics is to use Dirac's *braket* notation [Dirac 1958], in which a vector is written $|\psi\rangle$. This can be written as a column vector in the usual way:

$$|\psi\rangle = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} .$$

The notation $|\cdot\rangle$ is called a *ket*, and its dual $\langle\cdot|$ is called a *bra*. $\langle\psi|$ is the conjugate transpose (denoted †) of $|\psi\rangle$:

$$\langle\psi| = |\psi\rangle^\dagger = \begin{bmatrix} \alpha_1^* & \dots & \alpha_n^* \end{bmatrix}$$

where α^* is the complex conjugate of α.

A *Hilbert space* $\mathcal{H}$ is a *complete inner product space*; that is, a vector space with a zero element 0, and a unit element 1, such that all elements $|\psi\rangle, |\phi\rangle, |\varphi\rangle \in \mathcal{H}$ satisfy the axioms

$$\begin{aligned}
|\psi\rangle + |\phi\rangle &= |\phi\rangle + |\psi\rangle \\
(|\psi\rangle + |\phi\rangle) + |\varphi\rangle &= |\psi\rangle + (|\phi\rangle + |\varphi\rangle) \\
0 + |\psi\rangle &= |\psi\rangle \\
\alpha(\beta|\psi\rangle) &= (\alpha\beta)|\psi\rangle \\
(\alpha + \beta)|\psi\rangle &= \alpha|\psi\rangle + \beta|\psi\rangle \\
\alpha(|\psi\rangle + |\phi\rangle) &= \alpha|\psi\rangle + \alpha|\phi\rangle \\
1|\psi\rangle &= |\psi\rangle \\
\langle\psi|\phi\rangle &= \langle\phi|\psi\rangle^* \\
\langle\psi|\psi\rangle &\geq 0 \\
(\alpha\langle\psi| + \beta\langle\phi|)|\varphi\rangle &= \alpha\langle\psi|\varphi\rangle + \beta\langle\phi|\varphi\rangle
\end{aligned}$$

The *braket* $\langle\cdot|\cdot\rangle$ denotes the *inner product*. If

$$|\psi\rangle = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \text{ and } |\phi\rangle = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}$$

then the inner product is a complex number given by

$$\langle\psi|\phi\rangle = \alpha_1^*\beta_1 + \cdots + \alpha_n^*\beta_n \ .$$

2.1.2 Qubits

Associated with any physical system is a complex Hilbert space, called its *state space*. The system is completely described by a unit vector $|\psi\rangle$ within its state space, called the *state vector*. The system of primary interest to us is the *qubit* (or quantum bit). A qubit is a physical system with a 2-dimensional state space $\mathcal{H}^2$.

The set of vectors $\{|0\rangle, |1\rangle\}$ is called the *standard basis* of the qubit state space $\mathcal{H}^2$, where

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \ .$$

We can write the general state of a qubit as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.1}$$

where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. In contrast to a classical bit, whose state is either 0 or 1, the general state of a qubit is a linear combination, or *superposition*, of basis states.

The state space of a multiple qubit system is given by the *tensor product* ($\otimes$) of the individual systems' state spaces. This is in contrast to classical systems, in which the *Cartesian product* is used. The state space of an n-qubit system is a $2n$-dimensional complex Hilbert space; that is, the tensor product of n copies of $\mathcal{H}^2$:

$$\mathcal{H}^{2n} = \underbrace{\mathcal{H}^2 \otimes \cdots \otimes \mathcal{H}^2}_{n}$$

The standard basis of an n-qubit system is $\{|00\ldots0\rangle, |00\ldots1\rangle, \ldots, |11\ldots1\rangle\}$. We use the notation

$$|\psi\rangle \otimes |\phi\rangle = |\psi\rangle|\phi\rangle = |\psi\phi\rangle$$

to abbreviate the tensor product, thus $|00\ldots0\rangle = |0\rangle \otimes \cdots \otimes |0\rangle$. For example, a 2-qubit system has the standard basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ and a general state is given by the vector

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix}$$

where $\alpha, \beta, \gamma, \delta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$.

A quantum state is *separable* if it can be decomposed into the tensor product of two smaller systems. For example, the 2-qubit state $|\psi\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$ can be represented by the tensor product of two single qubit systems: $|\psi\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle)$. In vector notation this is

$$|\psi\rangle = \begin{bmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \otimes \begin{bmatrix} \gamma \\ \delta \end{bmatrix} .$$

A consequence of using the tensor product is that not all quantum states are separable. For example, the state $|\Phi^+\rangle = \alpha|00\rangle + \beta|11\rangle$ cannot be decomposed into two independent systems. Non-separable quantum states are called *entangled*.

2.1.3 Quantum Operators

A closed quantum system is one which is independent from the environment. The evolution of a closed system can be described by *unitary operations* on the quantum

state. A linear operator is *unitary* if $UU^\dagger = U^\dagger U = I$, where I is the identity operator. For a system that starts in state $|\psi\rangle$, after evolution described by the operator U, the state will be $|\phi\rangle = U|\psi\rangle$.

We now introduce the common quantum operators and give their matrix representations with respect to the standard basis. The *Pauli operators* are single qubit operators given by the matrix representations

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$
$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} .$$

In other texts, these are also referred to using the notations $(\sigma_0, \sigma_1, \sigma_2, \sigma_3)$ and $(I, \sigma_x, \sigma_y, \sigma_z)$. The other single qubit operators of interest are the *Hadamard*, *phase* and $\frac{\pi}{8}$ operators:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \qquad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} .$$

The effects of these operators on a general quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ are as follows:

$$\begin{aligned} I|\psi\rangle &= \alpha|0\rangle + \beta|1\rangle \\ X|\psi\rangle &= \alpha|1\rangle + \beta|0\rangle \\ Y|\psi\rangle &= i\alpha|0\rangle - i\beta|1\rangle \\ Z|\psi\rangle &= \alpha|0\rangle - \beta|1\rangle \\ H|\psi\rangle &= \frac{1}{\sqrt{2}}((\alpha+\beta)|0\rangle + (\alpha-\beta)|1\rangle) \\ S|\psi\rangle &= \alpha|0\rangle + i\beta|1\rangle \\ T|\psi\rangle &= \alpha|0\rangle + e^{i\pi/4}\beta|1\rangle . \end{aligned}$$

Of these operators, the Hadamard is particularly interesting because it can create and destroy superpositions. For example, $\mathsf{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $\mathsf{HH}|0\rangle = |0\rangle$. The final operator of importance is the 2-qubit *controlled-NOT* operator. It has the matrix representation

$$\mathsf{CNot} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} .$$

On basis states, it is defined by the map $|00\rangle \mapsto |00\rangle, |01\rangle, \mapsto |01\rangle, |10\rangle \mapsto |11\rangle, |11\rangle \mapsto |10\rangle$. The first qubit is considered the control, and depending on its state, the second qubit is 'flipped' accordingly.

The controlled-NOT (or CNot) operator, in conjunction with the Hadamard, is often used to create or destroy entanglement. Given a separable 2-qubit state $|\psi\rangle = |00\rangle$, we can apply the Hadamard operator to the first qubit, followed by the CNot operator to get:

$$\mathsf{CNot}.(\mathsf{H} \otimes \mathsf{I})|00\rangle = \mathsf{CNot}(\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \ .$$

Note that, by using the tensor product $(\mathsf{H} \otimes \mathsf{I})$, we have applied the (single-qubit) Hadamard operator to a 2-qubit state. Subscripts are often used to denote which qubits a gate is applied to, for example, $\mathsf{CNot}_{4,2}$ denotes the CNot gate applied to qubits 4 and 2. In these cases, the corresponding operator on the full state can be found by taking the tensor product with the identity operator on the other qubits. A change of basis may also be required.

2.1.4 Measurement

While a quantum system may be in the state $|\psi\rangle$, reading the state is achieved through measurement. Unlike the state of a classical bit which we can determine with certainty, Heisenberg's *uncertainty principle* tells us that it is not possible to learn the complete state of a quantum system. Instead quantum measurements produce a probabilistic outcome that is dependent on the state of the system.

We are primarily interested in the class of measurements known as *projective measurements*. Formally, projective measurements are described by a set of orthogonal *projection operators* $\{P_m\}$, where $P_m P_{m'} = \delta_{m,m'} P_m$, that act on the state space of the system. Projection operators are Hermitian, that is $P^\dagger = P$, and the index m refers to the possible measurement outcomes. For a system in state $|\psi\rangle$, the probability that the outcome of the measurement is m is given by

$$p(m) = \langle\psi|P_m|\psi\rangle$$

and the state after the measurement is

$$\frac{P_m|\psi\rangle}{\sqrt{\langle\psi|P_m|\psi\rangle}} \ .$$

The act of measurement forces the system into a particular state, and unlike unitary operators, is not reversible. It is therefore not possible to discover more information about the original state through multiple measurements; in fact, projective measure-

ments have the property that repeating them produces the same outcome and does not change the state. For this reason, measurement is considered *destructive*.

The measurement operators of a quantum measurement satisfy the completeness property

$$\sum_m P_m^\dagger P_m = I \ .$$

This corresponds to the condition that the probabilities sum to 1.

The type of measurements that we are are concerned with, are measurements with respect to the standard basis. In this basis, the measurement operators for a single qubit measurement are $P_0 = |0\rangle\langle 0|$ and $P_1 = |1\rangle\langle 1|$, corresponding to the two possible outcomes. The notation $|\psi\rangle\langle\phi|$ denotes the *outer product* of vectors $|\psi\rangle$ and $|\phi\rangle$. In vector notation,

$$P_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad P_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \ .$$

For a general state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the probability of obtaining the outcome 0 is

$$p(0) = \langle\psi|P_0|\psi\rangle = (\alpha^*\langle 0| + \beta^*\langle 1|)\alpha|0\rangle = |\alpha|^2 \ .$$

The corresponding state is then

$$\frac{P_0|\psi\rangle}{|\alpha|} = |0\rangle \ .$$

Similarly, for an outcome of 1, the probability is $p(1) = |\beta|^2$ and the resulting state is $|1\rangle$.

Measurement and entanglement have a particularly interesting connection; it is through measurement of entangled states that we can see the effects of non-locality. Let us consider the 2-qubit system in the entangled state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Measuring the first qubit corresponds to the measurement operators $P_0 = |0\rangle\langle 0| \otimes I$ and $P_1 = |1\rangle\langle 1| \otimes I$. This yields the following states and respective probabilities

$$|\psi_0\rangle = |00\rangle \text{ with probability } p(0) = \frac{1}{2}$$
$$|\psi_1\rangle = |11\rangle \text{ with probability } p(1) = \frac{1}{2} \ .$$

Now, if the outcome from this measurement was 0 and we measure the second qubit, we obtain, with probability 1, the outcome 0. Similarly, if the initial measurement gave 1, then the second measurement would give the outcome 1 with probability 1.

The state $|\Phi^+\rangle$ is one of four *maximally entangled* states, collectively known as the Bell states. They have the property that the measurement outcomes of the two

qubits are perfectly correlated. The other Bell states are

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$
$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$
$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \ .$$

2.1.5 Density Matrices

Density matrices (or *density operators*) provide an alternative approach to the representation of quantum states. The correspondence between state vectors and density matrices is not one-to-one; for each state vector there is a unique density matrix, but this mapping is not injective. Density matrices cannot represent the global phase of a state since, although the states $|\psi\rangle$ and $e^{i\theta}|\psi\rangle$ differ by the global phase coefficient $e^{i\theta}$, their respective density matrices are identical. In practice, it is not possible to determine the global phase of a state, hence $|\psi\rangle$ and $e^{i\theta}|\psi\rangle$ are indistinguishable. Moreover, it is not possible to distinguish any two states that have the same density matrix representation, even if their respective state vectors are not equal.

The correspondence of density matrices with physical indistinguishability makes this representation particularly useful when considering the observational properties of quantum systems. Aside from this, one of the main benefits of the density matrix representation is the ability to describe the state of individual subsystems of a composite system.

The density matrix (denoted ρ) of a system in the state $|\psi\rangle$ is given by the outer product

$$\rho = |\psi\rangle\langle\psi| \ .$$

For the general state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, this corresponds to the density matrix

$$\rho = (\alpha|0\rangle + \beta|1\rangle)(\alpha^*\langle 0| + \beta^*\langle 1|) = \begin{bmatrix}\alpha \\ \beta\end{bmatrix}\begin{bmatrix}\alpha^* & \beta^*\end{bmatrix} = \begin{bmatrix}|\alpha|^2 & \alpha\beta^* \\ \beta\alpha^* & |\beta|^2\end{bmatrix} \ .$$

State vectors represent *pure states* of a system. Density matrices are able to represent *ensembles of pure states* $\{p_i, |\psi_i\rangle\}$, in which the system is in one of the pure states $|\psi_i\rangle$, with respective probability p_i. The density matrix of this ensemble is given by

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \ .$$

For example, given the states $|0\rangle, |1\rangle$ with respective probabilities $|\alpha|^2, |\beta|^2$, the density

matrix of this ensemble is

$$\rho = |\alpha|^2|0\rangle\langle 0| + |\beta|^2|1\rangle\langle 1| = \begin{bmatrix} |\alpha|^2 & 0 \\ 0 & |\beta|^2 \end{bmatrix} .$$

Note that this is different than the superposition $|\psi\rangle$ given above, in which the state is a *combination* of $|0\rangle$ and $|1\rangle$.

The action of a unitary operator U on a density matrix ρ is given by $U\rho U^\dagger$ which, for a general state $|\psi\rangle$, is equivalent to $U|\psi\rangle\langle\psi|U^\dagger$. The state after a measurement with operators $\{P_m\}$ is given by

$$\rho_m = \frac{P_m \rho P_m^\dagger}{\mathrm{tr}(P_m^\dagger P_m \rho)}$$

where "tr" is the matrix *trace* operation. The derivation of the equality $p(m) = \mathrm{tr}(P_m^\dagger P_m \rho)$ can be found in the standard texts, e.g. [Nielsen and Chuang 2000, p 100].

Our main reason for using density matrices is to describe subsystems of composite systems. Given a composite system consisting of two subsystems A and B in a state $|\psi\rangle_{AB}$, it is not always possible to fully describe the respective states of the individual subsystems because they might not be separable. In these cases the subsystems can be described by their *reduced density matrices*.

The reduced density matrix of the subsystem A, denoted ρ^A, is given by

$$\rho^A = \mathrm{tr}_B(\rho^{AB})$$

where ρ^{AB} is the state of the composite system and tr_B is called the *partial trace* over B. This is often referred to as 'tracing out' or 'tracing over' B. If $|\psi\rangle = \sum_i |\psi_i\rangle_A |\phi_i\rangle_B$ then

$$\rho^A = \sum_{i,j} |\psi_i\rangle\langle\psi_j|_A \langle\phi_i|\phi_j\rangle_B .$$

For example, if a 2-qubit system is in the Bell state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ then

$$\rho = \frac{(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|)}{2} .$$

The reduced density matrix of the first qubit is then

$$\rho^A = \frac{|0\rangle\langle 0|\langle 0|0\rangle + |0\rangle\langle 1|\langle 0|1\rangle + |1\rangle\langle 0|\langle 1|0\rangle + |1\rangle\langle 1|\langle 1|1\rangle}{2} = \frac{|0\rangle\langle 0| + |1\rangle\langle 1|}{2} .$$

This is the same density matrix as for an ensemble of states $\{|0\rangle, |1\rangle\}$, each with probability 0.5. This equality represents the equivalence of observables in these systems.

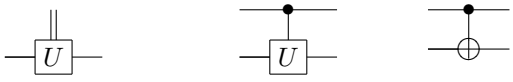
Two systems with the same density matrix may not have the same state (or ensemble of states), yet their measurement statistics will be identical.

2.1.6 Quantum Gates and Circuits

The quantum circuit model is a common way of representing quantum computations. It is analogous to classical logic circuits, replacing logic gates with quantum gates and classical wires with quantum wires.

Quantum circuits consist of parallel "quantum wires" running in a single direction, normally left to right. Each quantum wire corresponds to a single qubit. Unitary operators are represented by boxes —$\boxed{U}$— and appear on the quantum wire for the qubit(s) that they operate on. Measurements are represented by a "meter" symbol —$\boxed{\nearrow}$═, and measurement outcomes emerge on classical wires (double lines).

Unlike classical circuits, *fan-in* and *fan-out* are not permitted, and all quantum gates have the same number of input qubits as output qubits. Classical control of quantum gates is represented by a classical wire entering a quantum gate. Quantum controlled gates are represented by • on the control qubit, and the CNot gate marks the target with ⊕. These are illustrated as follows.



Classical control Controlled-U CNot

In the next section, we describe several quantum protocols using the circuit model.

2.2 Quantum Protocols

2.2.1 Quantum Teleportation

Quantum teleportation is a protocol which allows two users who share an entangled pair of qubits, to exchange an unknown quantum state by communicating only two classical bits. The protocol is illustrated in Figure 2.1 using the circuit model.

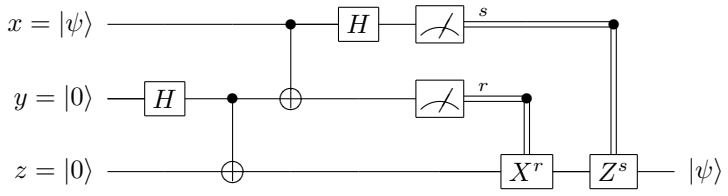
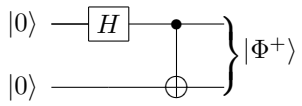


Figure 2.1. Quantum teleportation circuit.

The qubit labelled x is in some unknown state $|\psi\rangle$; this is the state to be "teleported". Following the usual convention, we call the sender Alice and the receiver Bob. Two qubits, y and z, must be prepared in the entangled state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. This is the achieved by applying a Hadamard and controlled-NOT operator to the state $|00\rangle$, represented by the following circuit:



Then qubit y is given to Alice and qubit z is given to Bob.

When Alice is ready to send the state, she applies the controlled-NOT operator to qubits x and y, followed by the Hadamard operator to qubit x. She then measures both of these qubits, and it is the outcomes of these measurements (values r and s) that she sends to Bob.

If we write $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ (representing an arbitrary quantum state, cf. (2.1)), then there are four possible outcomes for the measurements, each occurring with probability 0.25:

State	r	s	Correction
$\alpha\|000\rangle + \beta\|001\rangle$	0	0	I
$\alpha\|010\rangle - \beta\|011\rangle$	0	1	Z
$\alpha\|101\rangle + \beta\|100\rangle$	1	0	X
$\alpha\|111\rangle - \beta\|110\rangle$	1	1	ZX

Using the values r and s, Bob can determine which quantum operators to apply in order to "correct" the state; these are given in the above table. For example, if Bob receives the values 1,0, then he will apply the X operator to qubit z. After Bob has applied the appropriate corrections to his qubit, the state of z will be $|\psi\rangle$. Moreover, the state of Alice's qubit x will now be in the state $|0\rangle$ or $|1\rangle$, corresponding to the value s; this confirms that the quantum state has not been cloned.

2.2.2 Superdense Coding

Superdense coding is considered the opposite of teleportation, enabling two bits of classical information to be communicated be exchanging a single qubit. As with teleportation, superdense coding (also referred to as dense coding) is based on the two users sharing a pair of entangled qubits.

The superdense coding protocol is described by the circuit in Figure 2.2. The circuit model does not naturally represent implementation details such as the different users involved, however we have annotated the circuit in Figure 2.2 in order to identify the users in the protocol. The ability to describe these details, particularly

for communication protocols, is a significant benefit offered by process calculus and other formal specification languages.

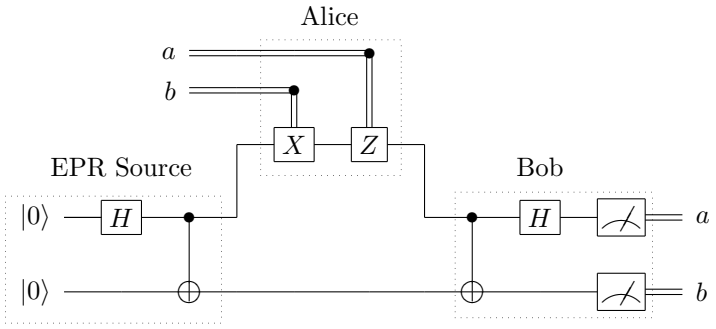


Figure 2.2. Superdense coding circuit.

Like teleportation, the superdense coding protocol also begins with the preparation of an entangled pair, labelled "EPR Source". Alice takes one qubit of the pair (x) and Bob takes the other (y). The bits to be transmitted are labelled a and b; the double lines represent classical wires. When Alice is ready to send, she applies a combination of the X and Z operators to qubit x depending on the values a and b. The appropriate encoding is as follows:

a	b	Operator	Resulting state
0	0	I	$\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$
0	1	X	$\frac{1}{\sqrt{2}}(\|10\rangle + \|01\rangle)$
1	0	Z	$\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$
1	1	ZX	$\frac{1}{\sqrt{2}}(\|01\rangle - \|10\rangle)$

Once Alice has performed this encoding, she sends her single qubit to Bob. Now that Bob has both qubits, he can determine which encoding Alice used, and therefore the corresponding values a and b. First, he applies a controlled-NOT operator to x and y, followed by the Hadamard applied to x; this results in the state $|ab\rangle$. He then measures both of these qubits to reveal the respective values. Because the state he measures is not a superposition, the outcome will be certain.

2.3 Process Calculus

Process calculus (also called process algebra) is an algebraic approach for the formal specification and verification of systems, usually involving concurrently executing and communicating components. Amongst the original and best-known process calculi are CCS [Milner 1982], ACP [Bergstra and Klop 1984] and CSP [Hoare 1978], from which many others have been inspired. Process calculus was initially developed to address

the verification of hardware systems, however further developments have lead to its application in software verification including object-oriented programming.

Process calculi have several features in common:

- **Syntax.** Models are built in a compositional manner using a small number of primitives and operators to combine these. This often includes sequential and parallel composition alongside operators for choice and scoping.
- **Semantics.** The syntax is usually accompanied by a structural operational semantics [Plotkin 1981] that defines the single-step execution capabilities of a system.
- **Behavioural equivalence.** Relations on processes are often defined as a means for equating systems which exhibit the "same behaviour". Notions of "refinement" can also be considered using preorders.

The typical process-algebraic approach to verification is to define two models; one which describes the design of the implemented system, and another (the specification) that describes the intended *high-level* behaviour. The correctness of the system with respect to the specification can then be established by using a notion of behavioural equivalence. This approach to verification will be used in Section 3.3.

2.3.1 Labelled Transition Systems

The operational semantics of a process calculus is normally defined by either a *reduction system* or a *labelled transition system (LTS)*. The first case uses a binary relation on terms called a *reduction*, and is able to represent the evolution of a process independently of its environment. The second case not only describes the evolution within a system, but also the interactions a system may have with the environment. The interactions that can occur are described by the *actions* that a process can perform. Being able to represent these interactions is important when considering the behaviour of communication systems hence we focus on the use of labelled transition systems instead of reductions in this thesis.

In a labelled transition system, the transition relations are labelled by *actions*, written $\xrightarrow{\alpha}$ where α is an action. Actions normally include *input*, *output* and *internal* action; we will use the respective notations $c?[x]$, $c![x]$ and τ for input on channel c, output on channel c, and internal action. The input and output actions are considered *visible* or *observable* actions.

The notation $P \xrightarrow{\alpha} Q$ means that P has the capability to perform action α, and after completing the action will reach a state where the remaining behaviour is Q. It may also be the case that *branching* may occur if $P \xrightarrow{\alpha_i} Q_i$ for a set of actions

$\{\alpha_i\}$ and set of states $\{Q_i\}$. The actions $\{\alpha_i\}$ are the *capabilities* of P. Transition relations are defined by inference rules of the form ([Plotkin 1981])

$$\frac{\text{premises}}{\text{conclusion}}\,.$$

For example, the following rules define actions ($\alpha.P$ performs action α then behaves as P), parallel execution ($P \parallel Q$ can behave as P without losing the capabilities of Q), and choice ($P + Q$ behaves as P or as Q):

$$\alpha.P \xrightarrow{\alpha} P$$

$$\frac{P \xrightarrow{\alpha} P'}{P \parallel Q \xrightarrow{\alpha} P' \parallel Q} \qquad \frac{Q \xrightarrow{\alpha} Q'}{P \parallel Q \xrightarrow{\alpha} P \parallel Q'}$$

$$\frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \qquad \frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$$

Rules for synchronisation or, in the case of value-passing calculi, communication enable parallel components to interact using complementary actions:

$$\frac{P \xrightarrow{c![v]} P' \quad Q \xrightarrow{c?[x]} Q'}{P \parallel Q \xrightarrow{\tau} P' \parallel Q'\{v/x\}}$$

Communication results from an output by one process and a corresponding input by another. In value-passing calculi, the output values are substituted in the receiving process (e.g. $Q'\{v/x\}$ denotes the substitution of value v in place of x). Communication is considered an internal action, and is therefore labelled τ.

These transition relations describe a directed graph in which the nodes represent the states and the edges represent the transitions. The edges emerging from a node indicate the *capabilities* of that state and the paths through the graph represent the possible executions.

Figure 2.3 illustrates the labelled transition system of a process $c?[x].d![x].\mathbf{0} \parallel c![0].\mathbf{0}$. This process consists of two components acting in parallel ($\parallel$) with each other; the branches of the transition system correspond to the possible interleavings of these two processes. The leftmost transition, labelled τ, corresponds to the synchronous input ($c?[x]$) and output ($c![0]$) of the two processes; this is action is called a *synchronisation* or *communication*. The other branches correspond to the external input ($c?[v]$) and output ($c![0]$) actions respectively. Each node is labelled with a process that describes the behaviour at the point. The leaves are labelled by the process $\mathbf{0}$, which has no action.

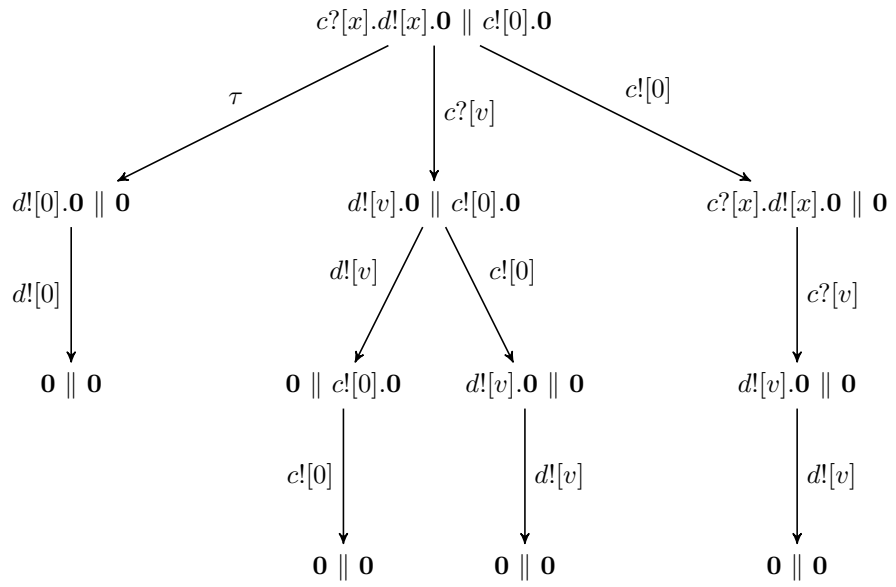


Figure 2.3. A labelled transition system.

2.3.2 Bisimulation

The notion of *bisimulation equivalence* was developed by Park [1981] and is based upon the concept of *simulation* due to Milner [1982]. If one process, *Sim*, *simulates* another, *Sys*, then it means that *Sim* is able to perform any action that *Sys* can, and that on completion of that action, the remaining behaviour of *Sim* can simulate the remaining behaviour of *Sys*. *Bisimulation* takes this concept a step further by requiring the simulation relation to be *symmetric*. If two process simulate one another then they are not necessarily bisimilar, since the two simulation relations may not be the symmetric forms of each other.

Strong Bisimilarity

There are many variants of bisimulation, of which *strong bisimulation* is the prototypical and most natural. Strong bisimulation requires every action to be matched exactly, regardless of whether they are internal or external actions.

Definition 2.1 (Strong Bisimulation)**.** A relation $\mathcal{R}$ is a *strong bisimulation* if whenever $(P, Q) \in \mathcal{R}$ then

- if $P \xrightarrow{\alpha} P'$ then $Q \xrightarrow{\alpha} Q'$ and $(P', Q') \in \mathcal{R}$,
- if $Q \xrightarrow{\alpha} Q'$ then $P \xrightarrow{\alpha} P'$ and $(P', Q') \in \mathcal{R}$.

This definition is often shortened by requiring the relation to be symmetric. This results in the following equivalent definition.

Definition 2.2 (Strong Bisimulation)**.** A symmetric relation $\mathcal{R}$ is a *strong bisimulation* if whenever $(P, Q) \in \mathcal{R}$ then if $P \xrightarrow{\alpha} P'$ then $Q \xrightarrow{\alpha} Q'$ and $(P', Q') \in \mathcal{R}$.

Strong bisimilarity is the union of all strong bisimulations. In other words, P and Q are strong bisimilar (denoted $P \sim Q$) if and only if there exists a strong bisimulation $\mathcal{R}$ such that $(P, Q) \in \mathcal{R}$.

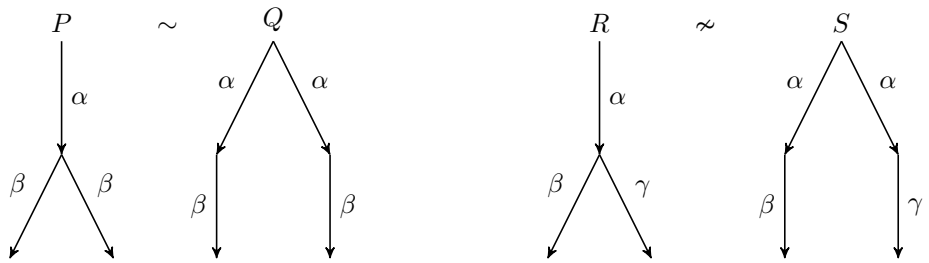


Figure 2.4. Strong bisimilarity.

Figure 2.4 gives examples of processes that are strong bisimilar ($P \sim Q$) and processes that are not strong bisimilar ($R \not\sim S$). If R makes the transition $\xrightarrow{\alpha}$, then the resulting process can perform either action β or γ. On the other hand, after making a transition $\xrightarrow{\alpha}$, the process S is in one of two states; one of which can perform action β, and the other which can perform action γ. Neither of these states are capable of both β and γ.

Weak Bisimilarity

One of the most useful properties of process calculus, particularly as an approach to verification, is the ability to abstract from the internal behaviour of a system. Strong bisimilarity requires processes to match execution on a step-by-step basis. This is too discriminating when comparing design and high-level specification because their respective internal behaviours are often very different. *Weak bisimilarity* identifies processes that exhibit the same external behaviour but allows internal actions to be matched by *zero or more* internal (τ) actions.

We introduce the notation $\Longrightarrow$ to be zero or more τ transitions, and $\overset{\alpha}{\Longrightarrow}$ is shorthand for $\Longrightarrow\xrightarrow{\alpha}$.

Definition 2.3 (Weak Bisimulation)**.** A symmetric relation $\mathcal{R}$ is a *weak bisimulation* if $(P, Q) \in \mathcal{R}$ implies that whenever $P \xrightarrow{\alpha} P'$ then there exists Q' such that $Q \overset{\alpha}{\Longrightarrow} Q'$ and $(P', Q') \in \mathcal{R}$.

Processes P and Q are *weak bisimilar*, denoted $P \approx Q$, if there exists a weak bisimulation $\mathcal{R}$ such that $(P, Q) \in \mathcal{R}$.

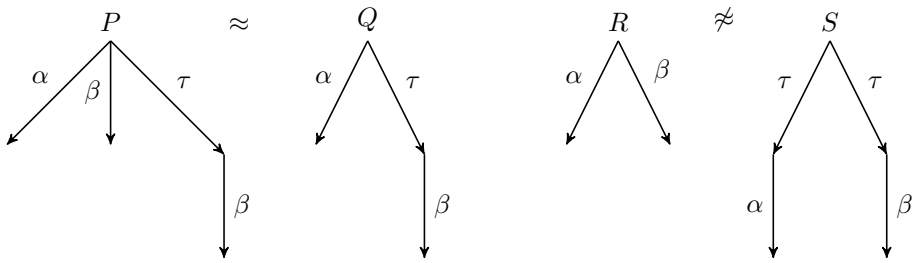


Figure 2.5. Weak bisimilarity.

Figure 2.5 shows examples of processes that are weak bisimilar ($P \approx Q$) and processes that are not ($R \not\approx S$). After making an internal transition, S loses one capability (either α or β), while R still has a choice between α and β. On the other hand, internal transitions by P and Q result in the inability to perform action α in both cases.

Branching Bisimilarity

Weak bisimilarity isn't the only equivalence that treats internal actions abstractly. *Branching bisimilarity* [van Glabbeek and Weijland 1996] is similar to weak bisimilarity, however it also matches the branching structure more accurately.

Definition 2.4 (Branching Bisimilarity)**.** A symmetric relation $\mathcal{R}$ is a *branching bisimulation* if $(P, Q) \in \mathcal{R}$ implies that whenever $P \xrightarrow{\alpha} P'$ then there exists Q' such that $Q \stackrel{\tau}{\Longrightarrow} Q' \xrightarrow{\alpha} Q''$ and $(P, Q') \in \mathcal{R}$ and $(P', Q'') \in \mathcal{R}$.

P and Q are *branching bisimilar* (denoted $P \leftrightarrows Q$) if there exists a branching bisimulation $\mathcal{R}$ such that $(P, Q) \in \mathcal{R}$.

The relationship of branching bisimilar processes is illustrated in Figure 2.6. The difference between this and weak bisimilarity, is the additional requirement that $P \leftrightarrows Q'$.

2.3.3 Quantum Process Calculus

In addition to *Communicating Quantum Processes (CQP)* which is used in this thesis, there are two other process calculi that have been designed to model quantum systems; *Quantum Process Algebra (QPAlg)* [Lalire 2006; Lalire and Jorrand 2004] and *Quantum CCS (qCCS)* [Feng et al. 2006; Ying et al. 2007, 2009]. Although there are

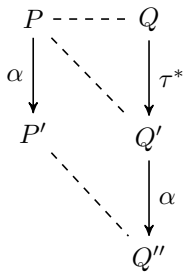


Figure 2.6. Branching bisimilarity.

significant differences between the languages and semantics, they each have common features not found in classical process calculi:

- **Quantum state.** The quantum state is considered as a global resource in order to represent entanglement.
- **Quantum communication.** It is possible for processes to send and receive both quantum and classical information.
- **Quantum operators.** Further primitives are included to model quantum operations such as unitary operators and measurements.

The transitions of quantum processes are dependent on the quantum state, and for this reason, the transition relations are defined using *configurations*. For example, a CQP configuration $(\sigma; \omega; P)$ consists of a quantum state σ and qubit list ω alongside a process P. A transition takes the form $(\sigma; \omega; P) \xrightarrow{\alpha} (\sigma'; \omega'; P')$. These transition relations must also deal with the probabilistic outcomes arising from quantum measurements; in CQP, there are probabilistic transitions which select one configuration from a probabilistic distribution:

$$p_1 \bullet (\sigma_1; \omega_1; P_1) \boxplus \cdots \boxplus \; p_n \bullet (\sigma_n; \omega_n; P_n) \overset{p_i}{\rightsquigarrow} (\sigma_i; \omega_i; P_i) \; .$$

A similar method is used in QPAlg, however in qCCS, probabilistic distributions are used throughout. The full syntax and semantics of CQP will be detailed in the following chapters. In particular, the semantics of measurement will be discussed in Chapters 3 and 4.

3

Behavioural Equivalence for Communicating Quantum Processes

This chapter presents a first attempt at using CQP for the verification of quantum protocols. We investigate behavioural equivalence for quantum processes, specifically *probabilistic branching bisimilarity*, in order to determine whether two systems act in the same way. Behavioural equivalence requires the ability to model the interaction of a process with the environment because it is these interactions that constitute the observational properties of the process. The existing reduction semantics of CQP describes only internal interactions, hence a core part of this chapter is dedicated to the definition of the operational semantics in terms of a labelled transition system.

There are a number of challenges involved in designing the labelled transition system when it comes to the representation of quantum information due to its non-local nature, and the related work by Feng et al. [2006]; Lalire [2005, 2006]; Ying et al. [2007, 2009] proves extremely valuable in this respect. It is important to note that the labelled transition system is not intended as a replacement for the reduction semantics, but provides a complementary semantics to use when modelling external interactions is required. We focus on the quantum teleportation protocol because it has a very simple high-level specification, yet the protocol features many aspects of the language, including measurement, unitary transformations and communication. With a specific protocol in mind, it is possible to critically consider the properties that an equivalence must possess, and the range of features used in teleportation significantly adds to the understanding we gain.

Although process equivalences already exist for QPAlg and qCCS, there are still

$$
\begin{array}{rcl}
T & ::= & \mathsf{Int} \mid \mathsf{Unit} \mid \mathsf{Qbit} \mid \widehat{\ }[\tilde{T}] \mid \mathsf{Op}(1) \mid \mathsf{Op}(2) \mid \cdots \\
v & ::= & \mathsf{0} \mid \mathsf{1} \mid \cdots \mid \mathsf{unit} \mid \mathsf{H} \mid \cdots \\
e & ::= & v \mid x \mid \mathsf{measure}\ \tilde{e} \mid \tilde{e} *\!= e^e \mid e + e \\
P & ::= & \mathbf{0} \mid (P \parallel P) \mid P + P \mid e?[\tilde{x}{:}\tilde{T}].P \mid e![\tilde{e}].P \mid \{e\}.P \mid (\mathsf{qbit}\ x)P \\
 & \mid & (\nu x{:}\widehat{\ }[T])P
\end{array}
$$

Figure 3.1. Syntax of CQP.

advantages to consider equivalence with CQP, including the use of the type system for the formal analysis of congruence properties. The implementation of the type system in CQP not only guarantees that qubits are only used by a single process, but also provides a clear and structured way to prove other results.

As we shall find, the equivalence we define in this chapter is not a congruence, however for a small class of quantum protocols this equivalence is preserved by all process constructs. This class of protocols includes quantum teleporatation, thereby allowing us to show the correctness of teleportation as a component in a larger system. Arguably the most important contribution from this chapter is the deeper understanding we gain about the observable aspects of quantum measurement. It is through the application to practical quantum processes that we are able to advance to the results in Chapter 4.

3.1 A Labelled Transition System for CQP

We begin this Section by introducing the syntax of CQP and the primitives for dealing with quantum information. The language was designed by Gay and Nagarajan [2005] based on the π-calculus [Milner 1999]. The π-calculus is a classical process calculus that extends value-passing CCS, with the inclusion of *channel mobility*; that is, channels can be communicated between processes, resulting in dynamic communication links. In this thesis, in order to simplify the presentation, we will omit channel mobility from the language.

CQP is fully motivated and described by Gay and Nagarajan [2005, 2006] along with the type system which is central to the proper treatment of quantum information. It is intended to be a flexible framework for studying quantum systems to which further features can be easily added, and maintaining this flexibility is an aim of this work. In Section 4.5 we consider some potential challenges involved in implementing some possible extensions.

$$
\begin{array}{rcl}
v & ::= & \ldots \mid q \mid c \\
E & ::= & [] \mid \mathsf{measure}\ E, \tilde{e} \mid \mathsf{measure}\ v, E, \tilde{e} \mid \ldots \mid \mathsf{measure}\ \tilde{v}, E \\
 & \mid & E + e \mid v + E \\
F & ::= & []?[\tilde{x}].P \mid []![\tilde{e}].P \mid v![[].\tilde{e}].P \mid v![v, [], \tilde{e}].P \mid \cdots \mid v![\tilde{v}, []].P \\
 & \mid & \{[]\}.P
\end{array}
$$

Figure 3.2. Internal syntax of CQP.

Syntax

The syntax of CQP is given by the grammar in Figure 3.1. We use the notation $\widetilde{e} = e_1, \ldots, e_n$ to denote a tuple, and write $|\widetilde{e}|$ for the length of a tuple. The syntax consists of types T, values v, expressions e, and processes P. Types include data types Int and Unit, the qubit type Qbit, channel types $\widehat{\ }[\widetilde{T}]$, and n-qubit unitary operators types $\mathsf{Op}(n)$; other data types may easily be included. Values include literal values of data types $\mathsf{0}, \mathsf{1}, \ldots$ and unit, and unitary operators such as the Hadamard operator H. Expressions include values, variables (x, y etc.), measurements $\mathsf{measure}\ \widetilde{e}$, the application of unitary operators $\widetilde{e} *= e^e$, and data operators such as $e + e'$. In contrast to [Gay and Nagarajan 2005, 2006], in which an informal notation σ_i was used in some examples to conditionally select a Pauli operator, we introduce an exponent on unitary operators to formally describe their conditional application; typically these exponents will be a measurement result, i.e. 0 or 1. The type system is used to ensure that, for example, a measurement or unitary operator is only applied to qubits.

Processes consist of the nil process $\mathbf{0}$, parallel compositions $P \parallel Q$, inputs $e?[\widetilde{x}{:}\widetilde{T}].P$, outputs $e![\widetilde{e}].P$, actions $\{e\}.P$, channel restriction $(\nu c{:}\widehat{\ }[\widetilde{T}])P$, and qubit declarations $(\mathsf{qbit}\ x)P$. We use the notation $\widetilde{x}{:}\widetilde{T} = x_1{:}T_1, \ldots, x_n{:}T_n$. Inputs and channel restrictions are often shortened to $c?[\widetilde{x}].P$ and $(\nu c)P$ where the respective types are obvious from the context, however their presence remains implicit.

For the purpose of investigating behavioural equivalence we have elected to simplify some features of the language. In particular, the language in [Gay and Nagarajan 2005] incorporated channel mobility based on its derivation from the π-calculus. Although this is arguably a useful feature, in this thesis we instead use restriction in the style of CCS to reduce the complexity of the notation and proofs. We replace the notation for channel declaration ($\mathsf{new}\ c$) used in [Gay and Nagarajan 2005] with channel restriction (νc) as used in CCS.

The *internal* syntax, defined by the grammar in Figure 3.2, extends the general syntax and is needed in order to define the operational semantics of CQP. Values are extended to include qubit names q which are generated at run-time. Evaluation contexts for expressions E and processes F are used to define the operational semantics

following the approach of Wright and Felleisen [1994]. E is defined recursively and leads to the left-to-right evaluation of expressions.

Given a process P we define its free variables $fv(P)$, free qubit names $fq(P)$ and free channel names $fc(P)$ as usual; the binders (of x or $\widetilde{x}$) are $y?[\widetilde{x}{:}\widetilde{T}]$ and $(\mathsf{qbit}\ x)$ and $(\nu x{:}\tilde{\frown}[\tilde{T}])$.

3.1.1 Describing External Interactions

The original presentation of CQP [Gay and Nagarajan 2005, 2006] defined the operational semantics using reductions, however this is not sufficient for describing the external interactions of a process. To consider behavioural equivalence it is necessary to define the operational semantics by a labelled transition system.

The main difference between the reduction semantics and the labelled transition system is the inclusion of input and output transitions. Indeed, when considering internal (τ) transitions alone then the two systems agree. The largest change is in the paradigm shift from considering closed quantum systems to open quantum systems (those which can interact with the environment). The use of closed systems by Gay and Nagarajan [2005, 2006] is motivated by the inability to fully describe the state of a quantum subsystem. This is coupled with the requirement to satisfy the no-cloning property, which is achieved by associating physical qubits with specific processes. In this interpretation there can be no qubits outside the system.

Although we must be able to describe interactions with the environment, including the input and output of qubits, there is still the question of how to represent the quantum state. One option is to consider the reduced density matrix of the system. This would be able to represent entanglement with the environment while only maintaining references to the qubits owned by the system. A significant drawback to such an approach is that if an entangled qubit is output to the environment and then later received, although the reduced density matrix can describe the system without that qubit, it is not possible to reconstruct the full state again on reception. The only viable solution to this is to keep qubits that have been output in the quantum state.

The approach taken by Lalire [2005, 2006] was to only allow input of unknown qubits, whilst output maintained the qubit in the known state. This suffered the drawback previously alluded to, and Feng et al. [2006] circumvented this problem by implementing two input rules; one allowing input of a known qubit and a second to allow input of an unknown qubit.

3.1.2 Semantics

Configurations

The execution of a system is not fully described by a process term, but also depends on the quantum state. For this reason, the operational semantics is defined using *configurations*, which represent both the quantum state and the process term. In [Gay and Nagarajan 2005, 2006] a configuration was a tuple $(\sigma; \Phi; P)$ where σ is a mapping from qubit names to the quantum state, Φ is a list of channel names, and P is a process. Because we are not considering π-calculus style channel mobility, the use of Φ is unnecessary. Instead we replace Φ with a list of qubit names ω, to give $(\sigma; \omega; P)$; this is needed for accounting purposes when dealing with external interactions.

Probabilistic distributions of configurations may also arise from quantum measurements; these are described by the probabilistic sum (denoted $\boxplus$) over configurations

$$p_1 \bullet (\sigma_1; \omega; P_1) \boxplus \cdots \boxplus \; p_n \bullet (\sigma_n; \omega; P_n)$$

in which $\sum_{1,\ldots,n} p_n = 1$. This is generally abbreviated to $\boxplus_{1,\ldots,n} \; p_i \bullet (\sigma_i; \omega; P_i)$.

For the evaluation of expressions we also introduce *expression configurations* $(\sigma; \omega; e)$; these are similar to configurations, but include an expression in place of the process. Expression configurations also have corresponding probabilistic distributions $\boxplus_{1,\ldots,n} \; p_i \bullet (\sigma_i; \omega; e_i)$.

Transition Relations

The evaluation of expressions is defined by the transition relations $\longrightarrow_v$ (on values) and $\longrightarrow_e$ (on expressions), given in Figure 3.3. Rules R-Plus, R-Measure and R-Trans deal with the evaluation of terms that result in values, including measurement which produces a probabilistic distribution over the possible measurement outcomes m, and unitary transformations which result in the literal unit. The most significant aspect of R-Measure and R-Trans is the effect they have on the quantum state. We note that the assignment of an integer m to the measurement outcomes follows the usual convention (for example, in a 2-qubit measurement the results $0, 1, 2, 3$ correspond to the states $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ respectively).

The labelled transition relation $\xrightarrow{\alpha}$ on configurations is defined by the rules in Figure 3.4. The internal transition rules ($\xrightarrow{\tau}$) correspond to the reduction rules in [Gay and Nagarajan 2005, 2006] with a few exceptions: L-Res replaces the channel declaration rule L-New to represent restriction, because we are not considering channel mobility; L-Com is defined inductively using L-In and L-Out in the standard way; L-Sum has been introduced to deal with non-deterministic choice.

The rules L-In and L-Out are additions to the semantics, representing the *visible*

$$(\sigma;\omega;u+v) \longrightarrow_v (\sigma;\omega;w) \text{ if } u \text{ and } v \text{ are integer literals and } w = u+v \quad \text{(R-Plus)}$$

$$([q_0,\ldots,q_{n-1} \mapsto \alpha_0|\phi_0\rangle + \cdots + \alpha_{2^n-1}|\phi_{2^n-1}\rangle];\omega;\mathsf{measure}\ q_0,\ldots,q_{r-1}) \longrightarrow_v \biguplus_{0\le m<2^r} p_m \bullet ([q_0,\ldots,q_{n-1} \mapsto \tfrac{\alpha_{l_m}}{p_m}|\phi_{l_m}\rangle + \cdots + \tfrac{\alpha_{u_m}}{p_m}|\phi_{u_m}\rangle];\omega;m) \quad \text{(R-Measure)}$$

where $l_m = 2^{n-r}m, u_m = 2^{n-r}(m+1)-1, p_m = |\alpha_{l_m}|^2 + \cdots + |\alpha_{u_m}|^2$

$$([q_0,\ldots,q_{n-1} \mapsto |\phi\rangle];\omega;q_0,\ldots,q_{r-1} \mathrel{*}= U^m) \longrightarrow_v ([q_0,\ldots,q_{n-1} \mapsto (U^m \otimes I_{n-r})|\phi\rangle];\omega;\mathsf{unit}) \quad \text{(R-Trans)}$$

$$\frac{(\sigma;\omega;e) \longrightarrow_v \biguplus_i p_i \bullet (\sigma_i;\omega;e_i)}{(\sigma;\omega;E[e]) \longrightarrow_e \biguplus_i p_i \bullet (\sigma_i;\omega;E[e_i])} \quad \text{(R-Context)}$$

Figure 3.3. Transition rules for values and expressions.

input and output actions respectively. Probabilistic transitions ($\overset{p_i}{\rightsquigarrow}$) are a special case of $\xrightarrow{\alpha}$, in which p_i is the probability of the transition.

Configurations are considered equivalent up to structural congruence, which is the smallest congruence relation on processes containing α-equivalence and closed under the rules in Figure 3.5.

3.1.3 Type System

The type system not only facilitates the combined use of quantum and classical data using a simple syntax, but also plays a central role in satisfying the no-cloning theorem of quantum information. The main results in this Section are *type preservation for* $\xrightarrow{\alpha}$ (Theorem 3.14) and the *unique ownership of qubits* (Theorem 3.15). Through these results it is guaranteed that quantum information is not duplicated. The type system is largely unchanged from [Gay and Nagarajan 2005, 2006], however due to the different semantics it is necessary to re-state and prove the results in the new setting.

The typing rules defined in Figure 3.6 apply to the syntax of processes given in Figure 3.1. Typing judgements are of two types: $\Gamma \vdash P$ means that P is well-typed in the environment Γ; and $\Gamma \vdash e : T$ means that an expression e has type T in the environment Γ. Typing environments are defined in the usual way as assignments of types to variables.

In order to prove results about executing processes, the *internal* type system is defined by the rules in Figure 3.7. The typing judgements are extended to include a list of qubits Σ, and therefore have the forms $\Gamma;\Sigma \vdash P$ and $\Gamma;\Sigma \vdash e : T$ for processes and expressions respectively. In contrast to [Gay and Nagarajan 2005, 2006], the list of channel names is no longer included.

$$\boxplus_i p_i \bullet (\sigma_i; \omega; P_i) \overset{p_i}{\rightsquigarrow} (\sigma_i; \omega; P_i) \qquad \text{(L-Prob)}$$

$$\frac{\widetilde{v} \text{ contains distinct qubit names } \widetilde{q}}{(\sigma; \omega, \widetilde{q}; c![\widetilde{v}].P) \xrightarrow{c![\widetilde{v}]} (\sigma; \omega; P)} \qquad \text{(L-Out)}$$

$$\frac{\widetilde{q} \text{ are distinct qubit names in } \widetilde{v}}{(\sigma; \omega; c?[\widetilde{x} : \widetilde{T}].P) \xrightarrow{c?[\widetilde{v}:\widetilde{T}]} (\sigma; \omega, \widetilde{q}; P\{\widetilde{v}/\widetilde{x}\})} \qquad \text{(L-In)}$$

$$\frac{(\sigma; \omega, \widetilde{q}; P) \xrightarrow{c![\widetilde{v}]} (\sigma; \omega; P') \qquad (\sigma; \omega; Q) \xrightarrow{c?[\widetilde{x}:\widetilde{T}]} (\sigma; \omega, \widetilde{q}; Q')}{(\sigma; \omega, \widetilde{q}; P \parallel Q) \xrightarrow{\tau} (\sigma; \omega, \widetilde{q}; P' \parallel Q')} \quad \text{if } |\widetilde{v}| = |\widetilde{x}| \qquad \text{(L-Com)}$$

$$\frac{(\sigma; \omega; P) \xrightarrow{\alpha} \boxplus_i p_i \bullet (\sigma_i; \omega; P_i)}{(\sigma; \omega; P + Q) \xrightarrow{\alpha} \boxplus_i p_i \bullet (\sigma_i; \omega; P_i)} \qquad \text{(L-Sum)}$$

$$\frac{(\sigma; \omega; P) \xrightarrow{\alpha} \boxplus_i p_i \bullet (\sigma_i; \omega; P_i)}{(\sigma; \omega; P \parallel Q) \xrightarrow{\alpha} \boxplus_i p_i \bullet (\sigma_i; \omega; P_i \parallel Q)} \qquad \text{(L-Par)}$$

$$\frac{(\sigma; \omega; P) \xrightarrow{\alpha} (\sigma'; \omega; P')}{(\sigma; \omega; (\nu c{:}\widehat{\ }[T]).P) \xrightarrow{\alpha} (\sigma'; \omega; (\nu c{:}\widehat{\ }[T]).P')} \quad \text{if } \alpha \notin \{c?[\cdot], c![\cdot]\} \qquad \text{(L-Res)}$$

$$([q_0, \ldots, q_n \mapsto |\psi\rangle]; \omega; (\mathsf{qbit}\ x)P) \xrightarrow{\tau} ([q_0, \ldots, q_n, q \mapsto |\psi\rangle|0\rangle]; \omega, q; P\{q/x\}) \quad \text{if } q \text{ is fresh} \qquad \text{(L-Qbit)}$$

$$(\sigma; \omega; \{v\}.P) \xrightarrow{\tau} (\sigma; \omega; P) \qquad \text{(L-Act)}$$

$$\frac{(\sigma; \omega; e) \longrightarrow_e \boxplus_i p_i \bullet (\sigma_i; \omega; e_i)}{(\sigma; \omega; F[e]) \xrightarrow{\tau} \boxplus_i p_i \bullet (\sigma_i; \omega; F[e_i])} \qquad \text{(L-Expr)}$$

Figure 3.4. Transition Relation Rules

The rules T-Par and IT-Par make use of the addition operator on environments (Definition 3.1). The purpose of this operator is to prevent the same qubit appearing in more than one process; it is the key to the proof of Theorem 3.15.

Definition 3.1 (Addition of Environments)**.** ([Gay and Nagarajan 2006, Definition 1]) The partial operation of adding a typed variable to an environment, $\Gamma + x{:}T$, is defined by

$$\Gamma + x{:}T = \begin{cases} \Gamma, x{:}T & \text{if } x \notin dom(\Gamma) \\ \Gamma & \text{if } T \neq \mathsf{Qbit} \text{ and } x{:}T \in \Gamma \\ \text{undefined} & \text{otherwise.} \end{cases}$$

This operation is extended inductively to a partial operation $\Gamma + \Delta$ on environments.

$$
\begin{array}{rcl}
P \parallel \mathbf{0} & \equiv & P \\
P \parallel Q & \equiv & Q \parallel P \\
P \parallel (Q \parallel R) & \equiv & (P \parallel Q) \parallel R \\
P + \mathbf{0} & \equiv & P \\
P + Q & \equiv & Q + P \\
P + (Q + R) & \equiv & (P + Q) + R \\
(\nu c)\mathbf{0} & \equiv & \mathbf{0} \\
(\nu c)(\nu d)P & \equiv & (\nu d)(\nu c)P \\
(\nu c)(P \parallel Q) & \equiv & P \parallel (\nu c)Q, \text{ if } c \notin fv(P)
\end{array}
$$

Figure 3.5. Rules for structural congruence.

$$\Gamma \vdash v : \mathsf{Int} \quad \text{if } v \text{ is an integer literal} \qquad \text{(T-IntLit)}$$

$$\Gamma \vdash \mathsf{unit} : \mathsf{Unit} \qquad \text{(T-Unit)}$$

$$\Gamma \vdash \mathsf{H} : \mathsf{Op}(2) \quad \text{etc.} \qquad \text{(T-Op)}$$

$$\Gamma, x{:}T \vdash x : T \qquad \text{(T-Var)}$$

$$\frac{\forall i(\Gamma \vdash x_i : \mathsf{Qbit}) \quad x_1, \ldots, x_n \text{ distinct}}{\Gamma \vdash \mathsf{measure}\ x_1, \ldots, x_n : \mathsf{Int}} \qquad \text{(T-Msure)}$$

$$\frac{\Gamma \vdash e : \mathsf{Int} \quad \Gamma \vdash e' : \mathsf{Int}}{\Gamma \vdash e + e' : \mathsf{Int}} \qquad \text{(T-Plus)}$$

$$\Gamma \vdash \mathbf{0} \qquad \text{(T-Nil)}$$

$$\frac{\Gamma, x{:}\mathsf{Qbit} \vdash P}{\Gamma \vdash (\mathsf{qbit}\ x)P} \qquad \text{(T-Qbit)}$$

$$\frac{\Gamma_1 \vdash P \quad \Gamma_2 \vdash Q \quad \Gamma_1 + \Gamma_2 \text{ defined}}{\Gamma_1 + \Gamma_2 \vdash P \parallel Q} \qquad \text{(T-Par)}$$

$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P + Q} \qquad \text{(T-Sum)}$$

$$\frac{\Gamma \vdash x : \hat{}[T_1, \ldots, T_n] \quad \Gamma, y_1{:}T_1, \ldots, y_n{:}T_n \vdash P}{\Gamma \vdash x?[y_1{:}T_1, \ldots, y_n{:}T_n].P} \qquad \text{(T-In)}$$

$$\frac{\begin{array}{c}\Gamma \vdash x : \hat{}[T_1, \ldots, T_m, \mathsf{Qbit}, \ldots, \mathsf{Qbit}] \quad \forall i.(T_i \neq \mathsf{Qbit}) \\ \forall i.(\Gamma \vdash e_i : T_i) \quad y_i \text{ distinct} \quad \Gamma \vdash P\end{array}}{\Gamma, y_1{:}\mathsf{Qbit} \ldots, y_n{:}\mathsf{Qbit} \vdash x![e_1, \ldots, e_m, y_1, \ldots, y_n].P} \qquad \text{(T-Out)}$$

$$\frac{\Gamma, x{:}\hat{}[T_1, \ldots, T_n] \vdash P}{\Gamma \vdash (\nu x{:}\hat{}[T_1, \ldots, T_n])P} \qquad \text{(T-Res)}$$

$$\frac{\Gamma \vdash e : T \quad \Gamma \vdash P}{\Gamma \vdash \{e\}.P} \qquad \text{(T-Act)}$$

$$\frac{\forall i(\Gamma \vdash x_i : \mathsf{Qbit}) \quad x_1 \ldots x_n \text{ distinct} \quad \Gamma \vdash U : \mathsf{Op}(n) \quad \Gamma \vdash e : \mathsf{Int} \quad \Gamma \vdash P}{\Gamma \vdash x_1, \ldots, x_n \mathrel{*}{=} U^e : \mathsf{Unit}} \qquad \text{(T-Trans)}$$

Figure 3.6. Typing rules.

$$\Gamma;\Sigma \vdash v : \mathsf{Int} \quad \text{if } v \text{ is an integer literal} \qquad \text{(IT-IntLit)}$$

$$\Gamma;\Sigma \vdash \mathsf{unit} : \mathsf{Unit} \qquad \text{(IT-Unit)}$$

$$\Gamma;\Sigma \vdash \mathbf{0} \qquad \text{(IT-Nil)}$$

$$\Gamma;\Sigma \vdash \mathsf{H} : \mathsf{Op}(2) \quad \text{etc.} \qquad \text{(IT-Op)}$$

$$\Gamma, x{:}T;\Sigma \vdash x : T \qquad \text{(IT-Var)}$$

$$\Gamma;\Sigma, q \vdash q : \mathsf{Qbit} \qquad \text{(IT-IdQ)}$$

$$\Gamma;\Sigma \vdash c : T \qquad \text{(IT-IdC)}$$

$$\frac{\Gamma;\Sigma \vdash e : \mathsf{Int} \quad \Gamma;\Sigma \vdash e' : \mathsf{Int}}{\Gamma;\Sigma \vdash e + e' : \mathsf{Int}} \qquad \text{(IT-Plus)}$$

$$\frac{\Gamma;\Sigma \vdash \widetilde{e} : \widetilde{\mathsf{Qbit}}}{\Gamma;\Sigma \vdash \mathsf{measure}\ \widetilde{e} : \mathsf{Int}} \qquad \text{(IT-Msure)}$$

$$\frac{\forall i.(\Gamma;\Sigma \vdash e_i : \mathsf{Qbit}) \quad \Gamma;\Sigma \vdash U : \mathsf{Op}(n) \quad \Gamma;\Sigma \vdash e : \mathsf{Int} \quad \text{each } e_i \text{ is either } x_i \text{ or } q_i\text{, all distinct}}{\Gamma;\Sigma \vdash e_1, \ldots, e_n \mathrel{*}= U^e : \mathsf{Unit}} \qquad \text{(IT-Trans)}$$

$$\frac{\Gamma, x{:}\mathsf{Qbit};\Sigma \vdash P}{\Gamma;\Sigma \vdash (\mathsf{qbit}\ x)P} \qquad \text{(IT-Qbit)}$$

$$\frac{\Gamma;\Sigma \vdash P \quad \Gamma;\Sigma \vdash Q}{\Gamma;\Sigma \vdash P + Q} \qquad \text{(IT-Sum)}$$

$$\frac{\Gamma_1;\Sigma_1 \vdash P \quad \Gamma_2;\Sigma_2 \vdash Q \quad \Gamma_1 + \Gamma_2 \text{ defined} \quad \Sigma_1 \cap \Sigma_2 = \emptyset}{\Gamma_1 + \Gamma_2;\Sigma_1 \cup \Sigma_2 \vdash P \parallel Q} \qquad \text{(IT-Par)}$$

$$\frac{\Gamma;\Sigma \vdash e : \widehat{}[T_1, \ldots, T_n] \quad \Gamma, y_1{:}T_1, \ldots, y_n{:}T_n;\Sigma \vdash P}{\Gamma;\Sigma \vdash e?[y_1{:}T_1, \ldots, y_n{:}T_n]\,.\,P} \qquad \text{(IT-In)}$$

$$\frac{\Gamma;\Sigma \vdash e : \widehat{}[\widetilde{T}, \widetilde{\mathsf{Qbit}}] \quad \forall i.(T_i \neq \mathsf{Qbit}) \quad \forall i.(\Gamma;\Sigma \vdash e_i : T_i) \quad \Gamma;\Sigma \vdash P \quad \widetilde{f} \text{ consists of distinct variables } \widetilde{f_x} \text{ and distinct qubit names } \widetilde{f_q}}{\Gamma, \widetilde{f_x}{:}\widetilde{\mathsf{Qbit}};\Sigma, \widetilde{f_q} \vdash e![e_1, \ldots, e_m, f_1, \ldots, f_n]\,.\,P} \qquad \text{(IT-Out)}$$

$$\frac{\Gamma;\Sigma \vdash e : T \quad \Gamma;\Sigma \vdash P}{\Gamma;\Sigma \vdash \{e\}\,.\,P} \qquad \text{(IT-Act)}$$

$$\frac{\Gamma, x{:}\widehat{}[T_1, \ldots, T_n];\Sigma \vdash P}{\Gamma;\Sigma \vdash (\nu x{:}\widehat{}[T_1, \ldots, T_n])P} \qquad \text{(IT-Res)}$$

Figure 3.7. Internal typing rules.

Soundness of the Type System

We now bring together the labelled transition system and the type system, and work towards proving Theorem 3.14 (type preservation) and Theorem 3.15 (unique ownership of qubits). These proofs follow the same course as in [Gay and Nagarajan 2006], with the significant difference being the inclusion of cases for L-In and L-Out in Theorem 3.14. The complete series of Lemmas leading up to this proof is included for completeness.

The addition of the qubit list ω to configurations has resulted in modifications to some of the following statements. The role of ω is to control the movement of qubit names when external interactions occur. The qubit list Σ used in typing judgements controls the internal movement of qubit names through rules such as IT-Par. Generally, for a configuration $(\sigma; \omega; P)$ we would have the typing judgment $\Gamma; \Sigma \vdash P$ and $\Sigma = \omega$. If P is a parallel composition $P_1 \parallel P_2$ then IT-Par gives the judgements $\Gamma_1; \Sigma_1 \vdash P_1$ and $\Gamma_2; \Sigma_2 \vdash P_2$ where $\Sigma_1 \cap \Sigma_2 = \emptyset$. Based on these judgements it is not possible to determine whether, for example, P_1 is able to input a qubit q. Such an input would only be possible if $q \notin \Sigma = \Sigma_1 \cup \Sigma_2$. It is because the typing environment for P_1 has no information about Σ_2 that it is necessary to use the global list ω.

In Lemma 3.3 (Type preservation for $\longrightarrow_v$) and Lemma 3.4 (Type preservation for $\longrightarrow_e$) we introduce the condition $\forall i.(\omega_i = \omega)$, to ensure that the qubit list is not affected by the evaluation of expressions. Because the qubit list can be changed by inputs and outputs, the condition in Theorem 3.14 is more elaborate; it requires any changes to ω to be reflected in changes to Σ.

Lemma 3.1 (Typability of Subterms in E)**.** *If $\mathcal{D}$ is a typing derivation concluding $\Gamma; \Sigma \vdash E[e] : T$ then there exists U such that $\mathcal{D}$ has a subderivation $\mathcal{D}'$ concluding $\Gamma; \Sigma \vdash e : U$ and the position of $\mathcal{D}'$ in $\mathcal{D}$ corresponds to the position of the hole in E.*

Proof. [Gay and Nagarajan 2006, Lemma 1] By induction on the structure of E. □

Lemma 3.2 (Replacement in E)**.** *If*

1. *$\mathcal{D}$ is a derivation concluding $\Gamma; \Sigma \vdash E[e] : T$,*
2. *$\mathcal{D}'$ is a subderivation of $\mathcal{D}$ concluding $\Gamma; \Sigma \vdash e : U$,*
3. *the position of $\mathcal{D}'$ in $\mathcal{D}$ matches the hole in E, and*
4. *$\Gamma; \Sigma \vdash e' : U$*

then $\Gamma; \Sigma \vdash E[e'] : T$.

Proof. [Gay and Nagarajan 2006, Lemma 2] Replace $\mathcal{D}'$ in $\mathcal{D}$ by a derivation of $\Gamma; \Sigma \vdash e' : U$. □

Lemma 3.3 (Type Preservation for $\longrightarrow_v$)**.** *If* $\Gamma;\Sigma \vdash e : T$ *and* $(\sigma;\omega;e) \longrightarrow_v \boxplus_i\ p_i \bullet (\sigma_i;\omega_i;e_i)$ *and* $\Sigma \subseteq \omega$ *and* $\omega \subseteq dom(\sigma)$ *then* $\forall i.(dom(\sigma_i) = dom(\sigma))$ *and* $\forall i.(\omega_i = \omega)$ *and* $\forall i.(\Gamma;\Sigma \vdash e_i : T)$*.*

Proof. Adaptation of [Gay and Nagarajan 2006, Lemma 3]. By case analysis of the derivation of $(\sigma;\omega;e) \longrightarrow_v \boxplus_i\ p_i \bullet (\sigma_i;\omega;e_i)$. □

Lemma 3.4 (Type Preservation for $\longrightarrow_e$)**.** *If* $\Gamma;\Sigma \vdash e : T$ *and* $(\sigma;\omega;e) \longrightarrow_e \boxplus_i\ p_i \bullet (\sigma_i;\omega_i;e_i)$ *and* $\Sigma \subseteq \omega$ *and* $\omega \subseteq dom(\sigma)$ *then* $\forall i.(dom(\sigma_i) = dom(\sigma))$ *and* $\forall i.(\omega_i = \omega)$ *and* $\forall i.(\Gamma;\Sigma \vdash e_i : T)$*.*

Proof. Adaptation of [Gay and Nagarajan 2006, Lemma 4]. The transition $(\sigma;\omega;e) \longrightarrow_e \boxplus_i\ p_i \bullet (\sigma_i;\omega_i;e_i)$ is derived from R-Context, so for some E we have $e = E[f]$, and $\forall i.(e_i = E[f_i])$ and $(\sigma;\omega;f) \longrightarrow_v \boxplus_i\ p_i \bullet (\sigma_i;\omega_i;f_i)$. From $\Gamma;\Sigma \vdash E[f] : T$, Lemma 3.1 gives $\Gamma;\Sigma \vdash f : U$ for some U, Lemma 3.3 gives $\forall i.(\Gamma;\Sigma \vdash f_i : U)$ and $\forall i.(dom(\sigma_i) = dom(\sigma))$ and $\forall i.(\omega_i = \omega)$, and Lemma 3.2 gives $\forall i.(\Gamma;\Sigma \vdash E[f_i] : T)$. □

Lemma 3.5 (Typability of Subterms in F)**.** *If* $\mathcal{D}$ *is a derivation concluding* $\Gamma;\Sigma \vdash F[e]$ *then there exists* T *such that* $\mathcal{D}$ *has a subderivation* $\mathcal{D}'$ *concluding* $\Gamma;\Sigma \vdash e : T$ *and the position of* $\mathcal{D}'$ *in* $\mathcal{D}$ *corresponds to the position of the hole in* F*.*

Proof. [Gay and Nagarajan 2006, Lemma 5] By induction on the structure of F. □

Lemma 3.6 (Replacement in F)**.** *If*

1. $\mathcal{D}$ *is a derivation concluding* $\Gamma;\Sigma \vdash F[e]$*,*
2. $\mathcal{D}'$ *is a subderivation of* $\mathcal{D}$ *concluding* $\Gamma;\Sigma \vdash e : T$*,*
3. *the position of* $\mathcal{D}'$ *in* $\mathcal{D}$ *matches the hole in* F*, and*
4. $\Gamma;\Sigma \vdash e' : T$

then $\Gamma;\Sigma \vdash F[e']$*.*

Proof. [Gay and Nagarajan 2006, Lemma 6] Replace $\mathcal{D}'$ in $\mathcal{D}$ by a derivation of $\Gamma;\Sigma \vdash e' : T$. □

Lemma 3.7 (Weakening for Expressions)**.** *If* $\Gamma;\Sigma \vdash e : T$ *and* $\Gamma \subseteq \Gamma'$ *and* $\Sigma \subseteq \Sigma'$ *then* $\Gamma';\Sigma' \vdash e : T$*.*

Proof. [Gay and Nagarajan 2006, Lemma 7] A straightforward induction on the derivation of $\Gamma;\Sigma \vdash e : T$. □

Lemma 3.8 (Weakening for Processes)**.** *If* $\Gamma;\Sigma \vdash P$ *and* $\Gamma \subseteq \Gamma'$ *and* $\Sigma \subseteq \Sigma'$ *then* $\Gamma';\Sigma' \vdash P$*.*

Proof. [Gay and Nagarajan 2006, Lemma 8] By induction on the derivation of $\Gamma; \Sigma \vdash P$. Most cases are straightforward; we present the most complex case, IT-Par.

IT-Par: We have the derivation

$$\frac{\Gamma_1; \Sigma_1 \vdash P \qquad \Gamma_2; \Sigma_2 \vdash Q}{\Gamma; \Sigma \vdash P \parallel Q}$$

where $\Gamma = \Gamma_1 + \Gamma_2$ and $\Sigma = \Sigma_1 \cup \Sigma_2$. If $\Gamma \subseteq \Gamma'$ then there exist Γ'_1 and Γ'_2 such that $\Gamma' = \Gamma'_1 + \Gamma'_2$ and $\Gamma_1 \subseteq \Gamma'_1$ and $\Gamma_2 \subseteq \Gamma'_2$ and $\Gamma' - \Gamma = (\Gamma'_1 - \Gamma_1) + (\Gamma'_2 - \Gamma_2)$. Similarly, if $\Sigma \subseteq \Sigma'$ there exist Σ'_1 and Σ'_2 such that $\Sigma' = \Sigma'_1 \cup \Sigma'_2$ and $\Sigma_1 \subseteq \Sigma'_1$ and $\Sigma_2 \subseteq \Sigma'_2$. Applying the inductive hypothesis gives $\Gamma'_1; \Sigma'_1 \vdash P$ and $\Gamma'_2; \Sigma'_2 \vdash Q$, therefore (by IT-Par) $\Gamma'; \Sigma' \vdash P \parallel Q$. □

Lemma 3.9. *If* $\Gamma; \Sigma \vdash e : T$ *then* $fv(e) \subseteq dom(\Gamma)$ *and* $fq(e) \subseteq \Sigma$.

Proof. Adaptation of [Gay and Nagarajan 2006, Lemma 9]. By induction on the derivation of $\Gamma; \Sigma \vdash e : T$. The base cases IT-Var and IT-IdQ are true by definition.

IT-Plus: $fv(e + e') = fv(e) \cup fv(e')$ and $fq(e + e') = fq(e) \cup fq(e')$. The inductive hypothesis gives $fv(e) \subseteq dom(\Gamma)$ and $fq(e) \subseteq \Sigma$ and $fv(e') \subseteq dom(\Gamma)$ and $fq(e') \subseteq \Sigma$. Therefore $fv(e + e') \subseteq dom(\Gamma)$ and $fq(e + e') \subseteq \Sigma$.

IT-Msure: Straightforward.

IT-Trans: Applying the inductive hypothesis gives $\forall i.(fv(e_i) \subseteq dom(\Gamma))$ and $\forall i.(fq(e_i) \subseteq \Sigma)$ and $fv(e) \subseteq dom(\Gamma)$ and $fq(e) \subseteq \Sigma$. Therefore $fv(\widetilde{e} *= U^e) \subseteq dom(\Gamma)$ and $fq(\widetilde{e} *= U^e) \subseteq \Sigma$. □

Lemma 3.10. *If* $\Gamma; \Sigma \vdash P$ *then* $fv(P) \subseteq dom(\Gamma)$ *and* $fq(P) \subseteq \Sigma$.

Proof. Adaptation of [Gay and Nagarajan 2006, Lemma 10]. By induction on the derivation of $\Gamma; \Sigma \vdash P$.

IT-Qbit: Applying the inductive hypothesis to the derivation of $\Gamma; \Sigma \vdash (\mathsf{qbit}\ x)P$ gives $fv(P) \subseteq dom(\Gamma, x{:}\mathsf{Qbit})$ and $fq(P) \subseteq \Sigma$. Therefore (by IT-Qbit) $fv((\mathsf{qbit}\ x)P) \subseteq dom(\Gamma, x{:}\mathsf{Qbit})$ and $qv((\mathsf{qbit}\ x)P) \subseteq \Sigma$. x is bound in $(\mathsf{qbit}\ x)P$, therefore $fv((\mathsf{qbit}\ x)P) \subseteq dom(\Gamma)$.

IT-Par: Applying the inductive hypothesis to the derivation of $\Gamma_1 + \Gamma_2; \Sigma_1 \cup \Sigma_2 \vdash P \parallel Q$ gives $fv(P) \subseteq dom(\Gamma_1)$ and $fq(P) \subseteq \Sigma_1$ and $fv(Q) \subseteq dom(\Gamma_2)$ and $fq(Q) \subseteq \Sigma_2$. Then $fv(P \parallel Q) = fv(P) \cup fv(Q) \subseteq dom(\Gamma_1) \cup dom(\Gamma_2) = dom(\Gamma)$ and $fq(P \parallel Q) = fq(P) \cup fq(Q) \subseteq \Sigma_1 \cup \Sigma_2 = \Sigma$.

IT-In: We have

$$\frac{\Gamma; \Sigma \vdash e : \widehat{}[\widetilde{T}] \qquad \Gamma, \widetilde{y}{:}\widetilde{T}; \Sigma \vdash P}{\Gamma; \Sigma \vdash e?[\widetilde{y}{:}\widetilde{T}].P}$$

By Lemma 3.9 $fv(e) \subseteq dom(\Gamma)$ and $fq(e) \subseteq \Sigma$. By the inductive hypothesis $fv(P) \subseteq dom(\Gamma, \widetilde{y}{:}\widetilde{T})$ and $fq(P) \subseteq \Sigma$. Each y_i is bound in $e?[\widetilde{y}{:}\widetilde{T}].P$, therefore $fv(e?[\widetilde{y}{:}\widetilde{T}].P) = fv(e) \cup fv(P) \subseteq dom(\Gamma)$ and $fq(e?[\widetilde{y}{:}\widetilde{T}].P) \subseteq \Sigma$.

IT-OUT: We have

$$\frac{\Gamma;\Sigma \vdash e : \hat{}[\widetilde{T}, \widetilde{\mathsf{Qbit}}] \quad \forall i.(\Gamma;\Sigma \vdash e_i : T_i) \quad \Gamma;\Sigma \vdash P \quad \forall i.(\Gamma, \widetilde{f}{:}\widetilde{\mathsf{Qbit}};\Sigma \vdash f : \mathsf{Qbit}) \quad \forall i.(\Gamma;\Sigma,\widetilde{q} \vdash q : \mathsf{Qbit})}{\Gamma, \widetilde{f}{:}\widetilde{\mathsf{Qbit}};\Sigma,\widetilde{q} \vdash e![\widetilde{e},\widetilde{f},\widetilde{q}].P}$$

Then by Lemma 3.9 and by the inductive hypothesis we have $fv(e) \subseteq dom(\Gamma)$ and $fq(e) \subseteq \Sigma$ and $\forall i.(fv(e_i) \subseteq dom(\Gamma)$ and $fq(e_i) \subseteq \Sigma)$ and $\forall i.(fv(f_i) \subseteq dom(\Gamma, \widetilde{f} : \widetilde{\mathsf{Qbit}})$ and $fq(f_i) \subseteq \Sigma)$ and $\forall i.(fv(q_i) \subseteq dom(\Gamma)$ and $fq(q_i) \subseteq \Sigma, \widetilde{q})$ and $fv(P) \subseteq dom(\Gamma)$ and $fq(P) \subseteq \Sigma$. Using Lemmas 3.7 and 3.8 we find $fv(e![\widetilde{e},\widetilde{f},\widetilde{q}].P) \subseteq dom(\Gamma, \widetilde{f}{:}\mathsf{Qbit})$ and $fq(e![\widetilde{e},\widetilde{f},\widetilde{q}].P) \subseteq \Sigma, \widetilde{q}$.

IT-ACT: Straightforward application of Lemma 3.9 and the inductive hypothesis.

IT-RES: Applying the inductive hypothesis gives $fv(P) \subseteq dom(\Gamma, x : \hat{}[\widetilde{T}])$ and $fq(P) \subseteq \Sigma$. x is bound in $(\nu x : \hat{}[\widetilde{T}])P$ therefore $fv((\nu x : \hat{}[\widetilde{T}])P) \subseteq dom(\Gamma)$ and $fq((\nu x{:}\hat{}[\widetilde{T}])P) \subseteq \Sigma$. □

Lemma 3.11 (Substitution in Expressions)**.** *Let $\widetilde{v}$ be values such that, for each i:*

1. *if $T_i \neq \mathsf{Qbit}$ then $\Gamma;\emptyset \vdash v_i : T_i$*
2. *if $T_i = \mathsf{Qbit}$ then v_i is q_i, a qubit name, such that $q_i \notin \Sigma$.*

Let $\widetilde{q}$ be the qubit names from $\widetilde{v}$ and assume they are distinct. Then $\Gamma, \widetilde{x}{:}\widetilde{T};\Sigma \vdash e : T$ if and only if $\Gamma;\Sigma,\widetilde{q} \vdash e\{\widetilde{v}/\widetilde{x}\} : T$.

Proof. Adaptation of [Gay and Nagarajan 2006, Lemma 11]. By induction on the derivation of $\Gamma;\Sigma \vdash e : T$. Reversing the argument proves the converse.

IT-VAR: If $T_i \neq \mathsf{Qbit}$ then by definition $\Gamma;\Sigma \vdash v_i : T_i$. If $T_i = \mathsf{Qbit}$ then $\Gamma;\Sigma, q_i \vdash q_i : \mathsf{Qbit}$.

IT-PLUS: Applying the inductive hypothesis to the derivation of $\Gamma, \widetilde{x}{:}\widetilde{T};\Sigma \vdash e + e' : \mathsf{Int}$ gives $\Gamma;\Sigma,\widetilde{q} \vdash e\{\widetilde{v}/\widetilde{x}\} : \mathsf{Int}$ and $\Gamma;\Sigma,\widetilde{q} \vdash e'\{\widetilde{v}/\widetilde{x}\} : \mathsf{Int}$. Therefore (by IT-PLUS) $\Gamma;\Sigma,\widetilde{q} \vdash (e + e')\{\widetilde{v}/\widetilde{x}\} : \mathsf{Int}$.

IT-MSURE: Applying the inductive hypothesis to the derivation of $\Gamma, \widetilde{x}{:}\widetilde{T};\Sigma \vdash \mathsf{measure}\ \widetilde{e} : \mathsf{Int}$ gives $\Gamma;\Sigma,\widetilde{q} \vdash \widetilde{e}\{\widetilde{v}/\widetilde{x}\} : \mathsf{Qbit}$. Therefore (by IT-MSURE) $\Gamma;\Sigma,\widetilde{q} \vdash (\mathsf{measure}\ \widetilde{e})\{\widetilde{v}/\widetilde{x}\} : \mathsf{Int}$.

IT-TRANS: Applying the inductive hypothesis to the derivation

$$\frac{\forall i.(\Gamma, \widetilde{x}{:}\widetilde{T};\Sigma \vdash e_i : \mathsf{Qbit}) \quad \Gamma, \widetilde{x}{:}\widetilde{T};\Sigma \vdash U : \mathsf{Op}(n) \quad \Gamma, \widetilde{x}{:}\widetilde{T};\Sigma \vdash e : \mathsf{Int}}{\Gamma, \widetilde{x}{:}\widetilde{T};\Sigma \vdash e_1, \ldots, e_n \mathrel{*}= U^e : \mathsf{Unit}}$$

gives $\forall i.(\Gamma;\Sigma,\widetilde{q} \vdash e_i\{\widetilde{v}/\widetilde{x}\} : \mathsf{Qbit})$ and $\Gamma;\Sigma,\widetilde{q} \vdash U\{\widetilde{v}/\widetilde{x}\} : \mathsf{Op}(n)$ and $\Gamma;\Sigma,\widetilde{q} \vdash e\{\widetilde{v}/\widetilde{x}\} : \mathsf{Int}$. Therefore (by IT-TRANS) $\Gamma;\Sigma,\widetilde{q} \vdash e_1, \ldots, e_n \mathrel{*}= (U^e)\{\widetilde{v}/\widetilde{x}\} : \mathsf{Unit}$. □

Lemma 3.12 (Substitution in Processes)**.** *Let $\widetilde{v}$ be values such that, for each i:*

1. *if* $T_i \neq \mathsf{Qbit}$ *then* $\Gamma; \emptyset \vdash v_i : T_i$
2. *if* $T_i = \mathsf{Qbit}$ *then* v_i *is* q_i*, a qubit name, such that* $q_i \notin \Sigma$.

Let $\widetilde{q}$ *be the qubit names from* $\widetilde{v}$ *and assume they are distinct. Then* $\Gamma, \widetilde{x}{:}\widetilde{T}; \Sigma \vdash P$ *if and only if* $\Gamma; \Sigma, \widetilde{q} \vdash P\{\widetilde{v}/\widetilde{x}\}$.

Proof. Adaptation of [Gay and Nagarajan 2006, Lemma 12]. By induction on the derivation of $\Gamma, \widetilde{x}{:}\widetilde{T}; \Sigma \vdash P$. The converse is proved by reversing the argument.

IT-Qbit: If

$$\frac{\Gamma, x : \mathsf{Qbit}, \widetilde{x}{:}\widetilde{T}; \Sigma \vdash P}{\Gamma, \widetilde{x}{:}\widetilde{T}; \Sigma \vdash (\mathsf{qbit}\ x)P}$$

then $x \notin \widetilde{x}$. Applying the inductive hypothesis gives $\Gamma, x : \mathsf{Qbit}; \Sigma, \widetilde{q} \vdash P\{\widetilde{v}/\widetilde{x}\}$. Therefore (by IT-Qbit) $\Gamma; \Sigma, \widetilde{q} \vdash (\mathsf{qbit}\ x)P\{\widetilde{v}/\widetilde{x}\}$.

IT-Par: We have the derivation

$$\frac{\Gamma_1; \Sigma_1 \vdash P \qquad \Gamma_2; \Sigma_2 \vdash Q}{\Gamma, \widetilde{x}{:}\widetilde{T}; \Sigma \vdash P \parallel Q}$$

where $\Gamma, \widetilde{x}{:}\widetilde{T} = \Gamma_1 + \Gamma_2$ and $\Sigma = \Sigma_1 \cup \Sigma_2$. Each variable of type Qbit is in either Γ_1 or Γ_2. The free variables of P and Q are contained in Γ_1 and Γ_2 respectively (Lemma 3.10), hence the substitution into $P \parallel Q$ splits into disjoint substitutions into P and Q. Applying the inductive hypothesis gives $\Gamma_1; \Sigma_1, \widetilde{q}_1 \vdash P\{\widetilde{v}/\widetilde{x}\}$ and $\Gamma_2; \Sigma_2, \widetilde{q}_2 \vdash Q\{\widetilde{v}/\widetilde{x}\}$. Then (by IT-Par) $\Gamma_1 + \Gamma_2; \Sigma, \widetilde{q} \vdash (P \parallel Q)\{\widetilde{v}/\widetilde{x}\}$.

IT-In: We have

$$\frac{\Gamma; \Sigma \vdash e : {}^\frown[\widetilde{T}] \qquad \Gamma, \widetilde{y}{:}\widetilde{T}; \Sigma \vdash P}{\Gamma; \Sigma \vdash e?[\widetilde{y}{:}\widetilde{T}].P}$$

Then by Lemma 3.11 $\Gamma; \Sigma, \widetilde{q} \vdash e\{\widetilde{v}/\widetilde{x}\} : {}^\frown[\widetilde{T}]$. Applying the inductive hypothesis gives $\Gamma, \widetilde{y}{:}\widetilde{T}; \Sigma, \widetilde{q} \vdash P\{\widetilde{v}/\widetilde{x}\}$. Note that $\widetilde{x}$ and $\widetilde{y}$ are distinct since the variables $\widetilde{y}$ are not free in $e?[\widetilde{y}{:}\widetilde{T}].P$. Then (by IT-In) $\Gamma; \Sigma, \widetilde{q} \vdash (e?[\widetilde{y}{:}\widetilde{T}].P)\{\widetilde{v}/\widetilde{x}\}$.

IT-Out: In the general case we have

$$\frac{\begin{array}{c}\Gamma, \widetilde{x_2} : \widetilde{\mathsf{Qbit}}, \widetilde{x_3} : \widetilde{T_3}; \Sigma \vdash e : {}^\frown[\widetilde{T}, \widetilde{\mathsf{Qbit}}] \\ \forall i.(\Gamma, \widetilde{x_2} : \widetilde{\mathsf{Qbit}}, \widetilde{x_3} : \widetilde{T_3}; \Sigma \vdash e_i : T_i, T_i \neq \mathsf{Qbit}) \quad \Gamma, \widetilde{x_2} : \widetilde{\mathsf{Qbit}}, \widetilde{x_3} : \widetilde{T_3}; \Sigma \vdash P\end{array}}{\Gamma, \widetilde{x_1} : \widetilde{\mathsf{Qbit}}, \widetilde{x_2} : \widetilde{\mathsf{Qbit}}, \widetilde{x_3} : \widetilde{T_3}, \widetilde{f} : \widetilde{\mathsf{Qbit}}; \Sigma, \widetilde{q_3} \vdash e![\widetilde{e}, \widetilde{x_1}, \widetilde{f}, \widetilde{q_3}].P}$$

where we are substituting $\widetilde{v} = \widetilde{q_1}\widetilde{q_2}\widetilde{v_3}$ for $\widetilde{x} : \widetilde{T} = \widetilde{x_1} : \widetilde{\mathsf{Qbit}}, \widetilde{x_2} : \widetilde{\mathsf{Qbit}}, \widetilde{x_3} : \widetilde{T_3}$ and each type in $\widetilde{T_3}$ is not Qbit and $\widetilde{q_1}, \widetilde{q_2} \notin \Sigma, \widetilde{q_3}$.

Then by Lemma 3.11 and the inductive hypothesis we get

$$\Gamma; \Sigma, \widetilde{q_2} \vdash e\{\widetilde{v}/\widetilde{x}\} : \widehat{\ }[\widetilde{T}, \widetilde{\mathsf{Qbit}}]$$
$$\forall i.(\Gamma; \Sigma, \widetilde{q_2} \vdash e_i\{\widetilde{v}/\widetilde{x}\} : T_i)$$
$$\Gamma; \Sigma, \widetilde{q_2} \vdash P\{\widetilde{v}/\widetilde{x}\}$$

Therefore (by IT-Out) $\Gamma, \widetilde{f} : \widetilde{\mathsf{Qbit}}; \Sigma, \widetilde{q_1}, \widetilde{q_2}, \widetilde{q_3} \vdash e\{\widetilde{v}/\widetilde{x}\}![\widetilde{e}\{\widetilde{v}/\widetilde{x}\}, \widetilde{q_1}, \widetilde{f}, \widetilde{q_3}].P\{\widetilde{v}/\widetilde{x}\}$. This corresponds to the substitution

$$\Gamma, \widetilde{f} : \widetilde{\mathsf{Qbit}}; \Sigma, \widetilde{q_1}, \widetilde{q_2}, \widetilde{q_3} \vdash (e![\widetilde{e}, \widetilde{x_1}, \widetilde{f}, \widetilde{q_3}].P)\{\widetilde{v}/\widetilde{x}\}$$

IT-Act: Applying Lemma 3.11 and the inductive hypothesis to the derivation of $\Gamma; \Sigma \vdash \{e\}.P$ gives $\Gamma; \Sigma, \widetilde{q} \vdash e\{\widetilde{v}/\widetilde{x}\} : T$ and $\Gamma; \Sigma, \widetilde{q} \vdash P$. Then (by IT-Act) $\Gamma; \Sigma, \widetilde{q} \vdash (\{e\}.P)\{\widetilde{v}/\widetilde{x}\}$.

IT-Res: We have

$$\frac{\Gamma, y{:}\widehat{\ }[\widetilde{T}]; \Sigma \vdash P}{\Gamma; \Sigma \vdash (\nu y{:}\widehat{\ }[\widetilde{T}])P}$$

$\widetilde{x}$ and $\widetilde{y}$ are distinct because $\widetilde{y}$ are bound variables. Applying the inductive hypothesis gives $\Gamma, y{:}\widehat{\ }[\widetilde{T}]; \Sigma, \widetilde{q} \vdash P\{\widetilde{v}/\widetilde{x}\}$. Then (by IT-Res) $\Gamma; \Sigma \vdash (\nu y{:}\widehat{\ }[\widetilde{T}])P\{\widetilde{v}/\widetilde{x}\}$. □

Lemma 3.13 (Structural Congruence Preserves Typing)**.** *If* $\Gamma; \Sigma \vdash P$ *and* $P \equiv Q$ *then* $\Gamma; \Sigma \vdash Q$.

Proof. Adaptation of [Gay and Nagarajan 2006, Lemma 13]. Straightforward induction on the derivation of $P \equiv Q$. □

Theorem 3.14 (Type Preservation for $\xrightarrow{\alpha}$)**.** *If* $\Gamma; \Sigma \vdash P$ *and* $(\sigma; \omega; P) \xrightarrow{\alpha} \boxplus_i\ p_i \bullet (\sigma_i; \omega'; P_i)$ *and* $\Sigma \subseteq \omega$ *and* $\omega \subseteq dom(\sigma)$ *then* $\forall i.(dom(\sigma) \subseteq dom(\sigma_i))$ *and there exists* Σ' *such that* $\Sigma' \subseteq \omega'$ *and* $\forall i.(\omega' \subseteq dom(\sigma_i))$ *and* $\forall i.(\Gamma; \Sigma' \vdash P_i)$ *and if* $\Sigma \subseteq \Sigma'$ *then* $\Sigma' - \Sigma = \omega' - \omega$*, or if* $\Sigma' \subset \Sigma$ *then* $\Sigma - \Sigma' = \omega - \omega'$.

Proof. Adaptation of [Gay and Nagarajan 2006, Theorem 1], accounting for the new rules L-In and L-Out. By induction on the derivation of $(\sigma; \omega; P) \xrightarrow{\alpha} \boxplus_i\ p_i \bullet (\sigma_i; \omega_i; P_i)$.

L-Expr: For some F we have $P = F[e]$ and $\forall i.(P_i = F[e_i])$ and $(\sigma; \omega; e) \longrightarrow_e \boxplus_i\ p_i \bullet (\sigma_i; \omega_i; e_i)$. From the derivation $\mathcal{D}$ of $\Gamma; \Sigma \vdash F[e]$, Lemma 3.5 gives T such that $\mathcal{D}'$ is a subderivation of $\mathcal{D}$ concluding $\Gamma; \Sigma \vdash e : T$. Lemma 3.4 gives $\forall i.(\Gamma; \Sigma \vdash e_i : T)$ and $\forall i.(dom(\sigma_i) = dom(\sigma))$ and $\forall i.(\omega_i = \omega)$, and Lemma 3.6 gives $\forall i(\Gamma; \Sigma \vdash F[e_i])$.

L-Out: Assume $\widetilde{v} = \widetilde{u}\widetilde{w}\widetilde{q}$ for variables $\widetilde{u}$, qubit variables $\widetilde{w}$ and qubit names $\widetilde{q}$, then we have $(\sigma; \omega, \widetilde{q}; c![\widetilde{v}].P) \xrightarrow{c![\widetilde{v}]} (\sigma; \omega; P)$ and

$$\frac{\Gamma; \Sigma \vdash c : \widehat{\ }[\widetilde{T}, \widetilde{\mathsf{Qbit}}] \qquad \Gamma; \Sigma \vdash P}{\Gamma, \widetilde{w} : \widetilde{\mathsf{Qbit}}; \Sigma, \widetilde{q} \vdash c![\widetilde{u}, \widetilde{w}, \widetilde{q}]\,.\,P}$$

Then $\Gamma;\Sigma \vdash P$ and $\Sigma \subseteq \omega$ since $\Sigma, \widetilde{q} \subseteq \omega, \widetilde{q}$ and $\Sigma, \widetilde{q} - \Sigma = \omega, \widetilde{q} - \omega = \widetilde{q}$ and $dom(\sigma)$ is unchanged.

L-In: We have $(\sigma;\omega;c?[\widetilde{x}{:}\widetilde{T},\widetilde{y}{:}\widetilde{\mathsf{Qbit}}].P) \xrightarrow{c?[\widetilde{u}\widetilde{q}]} (\sigma;\omega';P\{\widetilde{v}\widetilde{q}/\widetilde{x}\widetilde{y}\})$ where we are substituting $\widetilde{v} = \widetilde{u}\widetilde{q}$ for $\widetilde{x}\widetilde{y}$ using non-qubit values $\widetilde{u}$ and distinct qubit names $\widetilde{q}$. Then $\omega' = \omega, \widetilde{q}$ and

$$\frac{\Gamma;\Sigma \vdash c : \widehat{\ }[\widetilde{T}, \widetilde{\mathsf{Qbit}}] \quad \Gamma, \widetilde{x}{:}\widetilde{T}, \widetilde{y}{:}\widetilde{\mathsf{Qbit}};\Sigma \vdash P}{\Gamma;\Sigma \vdash c?[\widetilde{x}{:}\widetilde{T},\widetilde{y}{:}\widetilde{\mathsf{Qbit}}]\,.\,P}$$

where $\forall i.(T_i \neq \mathsf{Qbit})$. By Lemma 3.12 we have $\Gamma;\Sigma,\widetilde{q} \vdash P\{\widetilde{u}\widetilde{q}/\widetilde{x}\widetilde{y}\}$. $\Sigma,\widetilde{q} \subseteq \omega,\widetilde{q}$ and $\Sigma,\widetilde{q} - \Sigma = \omega,\widetilde{q} - \omega = \widetilde{q}$ and $dom(\sigma)$ is constant. We have $\widetilde{q} \subseteq dom(\sigma)$ and $\omega \subseteq dom(\sigma)$, therefore $\omega,\widetilde{q} \subseteq dom(\sigma)$.

L-Com: We have

$$\frac{(\sigma;\omega;P) \xrightarrow{c![\widetilde{v}]} (\sigma;\omega,\widetilde{q};P') \quad (\sigma;\omega,\widetilde{q};Q) \xrightarrow{c?[\widetilde{v}]} (\sigma;\omega;Q')}{(\sigma;\omega,\widetilde{q};P \parallel Q) \xrightarrow{\tau} (\sigma;\omega,\widetilde{q};P' \parallel Q')}$$

where $\widetilde{q}$ are the distinct qubit names in $\widetilde{v}$. The typing derivation is

$$\frac{\Gamma_1;\Sigma_1 \vdash P \quad \Gamma_2;\Sigma_2 \vdash Q \quad \Gamma_1 + \Gamma_2 \text{ defined} \quad \Sigma_1 \cap \Sigma_2 = \emptyset}{\Gamma_1 + \Gamma_2;\Sigma_1 \cup \Sigma_2 \vdash P \parallel Q}$$

Applying the inductive hypothesis gives $\Sigma_1 \subseteq \omega$ and $\Sigma_2 \subseteq \omega,\widetilde{q}$ and Σ_1', Σ_2' such that $\Sigma_1' \subseteq \omega,\widetilde{q}$ and $\Sigma_2' \subseteq \widetilde{q}$ and $\Sigma_1' - \Sigma_1 = \widetilde{q}$ and $\Sigma_2 - \Sigma_2' = \widetilde{q}$ and $\omega,\widetilde{q} \subseteq dom(\sigma)$ and $\Gamma_1;\Sigma_1' \vdash P'$ and $\Gamma_2;\Sigma_2' \vdash Q'$. Then $\Sigma_1' \cup \Sigma_2' = \Sigma_1 \cup \Sigma_2$ and $\Sigma_1' \cap \Sigma_2' = (\Sigma_1 \cup \widetilde{q}) \cap (\Sigma_2 - \widetilde{q}) = (\Sigma_1 \cap (\Sigma_2 - \widetilde{q})) \cup (\widetilde{q} \cap (\Sigma_2 - \widetilde{q})) \subseteq \Sigma_1 \cap \Sigma_2 = \emptyset$. Then by using IT-Par we obtain $\Gamma_1 + \Gamma_2;\Sigma_1' \cup \Sigma_2' \vdash P' \parallel Q'$.

L-Act: We have the transition $(\sigma;\omega;\{v\}.P) \xrightarrow{\tau} (\sigma';\omega;P)$. The typing derivation concluding $\Gamma;\Sigma \vdash \{v\}.P$ contains the required hypothesis $\Gamma;\Sigma \vdash P$. Γ, Σ and ω are unchanged.

L-Res: The transition has the derivation

$$\frac{(\sigma;\omega;P) \xrightarrow{\alpha} (\sigma';\omega';P')}{(\sigma;\omega;(\nu c{:}\widehat{\ }[\widetilde{T}])P) \xrightarrow{\alpha} (\sigma';\omega';(\nu c{:}\widehat{\ }[\widetilde{T}])P')}$$

The typing derivation has the hypothesis $\Gamma, x{:}\widehat{\ }[\widetilde{T}];\Sigma \vdash P$. Applying the inductive hypothesis gives $\Gamma, x{:}\widehat{\ }[\widetilde{T}];\Sigma' \vdash P'$ where $\Sigma' \subseteq \omega'$ and $\omega' \subseteq dom(\sigma')$ and $dom(\sigma) \subseteq dom(\sigma')$ and if $\Sigma \subseteq \Sigma'$ then $\Sigma' - \Sigma = \omega' - \omega$ and if $\Sigma' \subseteq \Sigma$ then $\Sigma - \Sigma' = \omega - \omega'$. Therefore (by IT-Res) $\Gamma';\Sigma' \vdash (\nu c{:}\widehat{\ }[\widetilde{T}])P'$.

L-Qbit: We have $(\sigma;\omega;(\mathsf{qbit}\ x)P) \xrightarrow{\tau} (\sigma';\omega,q;P\{q/x\})$ where q is fresh and

$$\frac{\Gamma, x{:}\mathsf{Qbit};\Sigma \vdash P}{\Gamma;\Sigma \vdash (\mathsf{qbit}\ x)P}$$

Applying the inductive hypothesis and Lemma 3.12 gives the required judgement $\Gamma; \Sigma, q \vdash P\{q/x\}$. We have $\Sigma, \widetilde{q} \subseteq \omega, \widetilde{q}$ and $\Sigma, \widetilde{q} - \Sigma = \omega, \widetilde{q} - \omega$ and $dom(\sigma) \subseteq dom(\sigma')$ and $\omega, \widetilde{q} \subseteq dom(\sigma')$.

L-SUM: We have

$$\frac{(\sigma; \omega; P) \xrightarrow{\alpha} \boxplus_i \, p_i \bullet (\sigma_i; \omega_i; P_i)}{(\sigma; \omega; P + Q) \xrightarrow{\alpha} \boxplus_i \, p_i \bullet (\sigma_i; \omega_i; P_i)}$$

and

$$\frac{\Gamma; \Sigma \vdash P \qquad \Gamma; \Sigma \vdash Q}{\Gamma; \Sigma \vdash P + Q}$$

Applying the inductive hypothesis gives Σ' such that $\forall i.(\Sigma' \subseteq \omega_i)$ and $\forall i.(dom(\sigma) \subseteq dom(\sigma_i))$ and $\forall i.(\omega_i \subseteq \sigma_i)$ and $\forall i.(\Gamma; \Sigma' \vdash P_i)$ and if $\Sigma \subseteq \Sigma'$ then $\Sigma' - \Sigma = \omega' - \omega$ and if $\Sigma' \subset \Sigma$ then $\Sigma - \Sigma' = \omega - \omega'$.

L-PAR: The transition $(\sigma; \omega; P \parallel Q) \xrightarrow{\alpha} \boxplus_i \, p_i \bullet (\sigma_i; \omega'; P_i \parallel Q)$ has the hypothesis $(\sigma; \omega; P) \xrightarrow{\alpha} \boxplus_i \, p_i \bullet (\sigma_i; \omega'; P_i)$. We have the type derivation

$$\frac{\Gamma_1; \Sigma_1 \vdash P \qquad \Gamma_2; \Sigma_2 \vdash Q \qquad \Gamma_1 + \Gamma_2 \text{ defined} \qquad \Sigma_1 \cap \Sigma_2 = \emptyset}{\Gamma_1 + \Gamma_2; \Sigma_1 \cup \Sigma_2 \vdash P \parallel Q}$$

We are given that $\Sigma_1 \cup \Sigma_2 \subseteq \omega$, hence $\Sigma_1 \subseteq \omega$. Applying the inductive hypothesis gives Σ_1' such that $\forall i.(\Gamma_1; \Sigma_1' \vdash P_i)$ and $\Sigma_1' \subseteq \omega'$ and $\forall i.(dom(\sigma) \subseteq dom(\sigma_i))$ and $\forall i.(\omega \subseteq \sigma_i)$ and if $\Sigma \subseteq \Sigma'$ then $\Sigma' - \Sigma = \omega' - \omega$ and if $\Sigma' \subset \Sigma$ then $\Sigma - \Sigma' = \omega - \omega'$.

If $\Sigma_1 \subseteq \Sigma_1'$ then $\Sigma_1' \cap \Sigma_2 = (\Sigma_1 \cup (\omega' - \omega)) \cap \Sigma_2 = (\Sigma_1 \cap \Sigma_2) \cup (\omega' - \omega \cap \Sigma_2) = \emptyset$ because $\Sigma_2 \subseteq \omega$. If $\Sigma_1' \subset \Sigma$ then $\Sigma_1' \cap \Sigma_2 \subseteq \Sigma_1 \cap \Sigma_2 = \emptyset$. Therefore (by IT-PAR), for each i we obtain $\Gamma_1 + \Gamma_2; \Sigma_1' \cup \Sigma_2 \vdash P_i \parallel Q$. □

Theorem 3.15 (Unique Ownership of Qubits)**.** *If* $\Gamma; \Sigma \vdash P \parallel Q$ *then* $fq(P) \cap fq(Q) = \emptyset$.

Proof. [Gay and Nagarajan 2006, Theorem 2] The final step in the derivation of $\Gamma; \Sigma \vdash P \parallel Q$ has the form

$$\frac{\Gamma_1; \Sigma_1 \vdash P \quad \Gamma_2; \Sigma_2 \vdash Q \quad \Gamma_1 + \Gamma_2 \text{ defined} \quad \Sigma_1 \cap \Sigma_2 = \emptyset}{\Gamma; \Sigma \vdash P \parallel Q}$$

where $\Gamma = \Gamma_1 + \Gamma_2$ and $\Sigma = \Sigma_1 \cup \Sigma_2$. By Lemma 3.10, $fq(P) \subseteq \Sigma_1$ and $fq(Q) \subseteq \Sigma_2$. Since $\Sigma_1 \cap \Sigma_2 = \emptyset$ we have $fq(P) \cap fq(Q) = \emptyset$. □

3.2 Quantum Process Equivalence

In the previous section, we established the semantics that enable us to describe external interactions of quantum processes. We now consider the observational equivalence

of processes based on this semantics. In Section 2.3.2, we introduced bisimulation as a notion of process equivalence. We now consider the adaptation of bisimulation for CQP processes.

Bisimulation centers around the matching of *actions* that a process can perform. Internal behaviour is straightforward to match since these transitions are always labelled by τ. However when it comes to external actions, there is more to the label than just input and output. For example, the action $c![v, q]$ indicates an *output* on *channel name* c of a *value* v and a *qubit name* q. Matching actions, channel names and values is straightforward because there is no other associated information, unlike the qubit name, which has an associated state. This gives us the option of either matching the qubit name or the qubit state, and to make this choice, we must consider the behaviour that we wish to capture. We briefly digress, and consider how quantum information fits into the general picture of equivalence.

In terms of quantum information, we can consider a system in two ways: As a function from an initial quantum state to a final state; or as a process that receives and transmits information in the course of execution. The former approach is used by Feng et al. [2007]; it is necessary to match the physical qubits (the qubit names) at each output, and to match the final quantum state (when there are no further transitions). A similar approach is followed by Ying et al. [2007, 2009] in which, at each execution step, the state transformations are matched. The approximate versions of their equivalences require the transformations to match within a specified accuracy, according to the *diamond distance* measure on *superoperators*.

On the other hand, Lalire [2005, 2006] matches the quantum state of a qubit when it appears in an output action. We must remember that, due to entanglement it does not necessarily make sense to talk about the "state" of a particular qubit; instead the reduced density matrix is used, which enables the comparison of non-separable subsystem states up to observational indistinguishability. This approach has the advantage of combining observations of action and quantum information, which is suited to communication protocols involving many external interactions. In contrast, the alternate approach is arguably more suited to computation in which the final state is more important.

Let us consider how these treatments relate to communication protocols; we use quantum teleportation as an example. In the teleportation protocol there is a single qubit which we consider as the input to the system. During execution, two ancilla qubits are used for the internal communication between the two parties. At the end of execution there is one qubit of interest; that which is now in the same state as the initial input. This "output" qubit is not the same physical qubit as the initial input, and the two ancilla qubits end up in one of the four 2-qubit basis states, depending on the measurement results that were obtained during execution. To capture the

behaviour of teleportation, that "the quantum state is transferred from one qubit to another", means we should concentrate on the quantum state instead of the physical qubits, and that we are interested in specific "output" qubits instead of the global quantum state. By modelling the teleportation protocol in such a way that describes a system implementation of the protocol, we can identify the input and output qubits, thereby enabling us to ignore ancilla qubits and their states.

Teleportation is just one example of a communication protocol, but it highlights our interest in the *interactions* as opposed to the end result. Therefore, we choose to match output actions containing qubit names by equating the respective reduced density matrices. For example, the actions $c![q]$ and $c![r]$ would match if the reduced density matrices of q and r are equal.

3.2.1 Probabilistic Branching Bisimulation

The transition relation $\xrightarrow{\alpha}$ is defined on configurations, and therefore we must first consider an equivalence on configurations before extending this to an equivalence on processes. We combine the notion of branching bisimulation, which is a weak relation that preserves branching structure, with matching for probabilistic transitions. The resulting *probabilistic branching bisimulation* is similar to the equivalence defined by Lalire [2006], however a significant difference is in our treatment of non-determinism. For the purposes of simplification, non-deterministic branching was modelled as equiprobable choice by Lalire. However, a significant drawback of this approach is that it is not preserved by parallel composition.

A common method to account for non-determinism is the use of *schedulers* or *adversaries*, which assign probabilities to executions. Equivalence is then based on the existence of specific schedulers or adversaries, and this approach was used by Feng et al. [2007]. To avoid the added complexity of introducing schedulers, we follow a similar approach to Lalire [2006], and define a function μ, which assigns a probability to each transition. However, we maintain a distinction between non-deterministic and probabilistic branching, and use a function that is preserved by parallel composition. Our choice of function is based on the bisimulation by Trčka and Georgievska [2008], which assigns probability 1 to all non-deterministic transitions. The CQP transition system fits into the alternating class of probabilistic transition systems, because all probabilistic configurations result in a probabilistic transition to a non-probabilistic configuration. As a result, this probabilistic function is simpler than a corresponding function for a non-alternating system, which would have to account for sequences of consecutive probabilistic transitions.

Let $\mathcal{S}$ be the set of configurations. The relations $\longrightarrow$ and $\overset{p}{\rightsquigarrow}$ induce a partitioning of $\mathcal{S}$ into non-deterministic configurations $\mathcal{S}_n$ and probabilistic configurations $\mathcal{S}_p$: Let $\mathcal{S}_p = \{s \in \mathcal{S} \mid \exists p \in (0,1], \exists t \in \mathcal{S}, s \overset{p}{\rightsquigarrow} t\}$; and let $\mathcal{S}_n = \mathcal{S} \setminus \mathcal{S}_p$. By this definition a

configuration with no transitions belongs to $\mathcal{S}_n$.

We now define the probabilistic function $\mu : \mathcal{S} \times \mathcal{S} \to [0,1]$ in the style of Trčka and Georgievska [2008]:

$$\mu(s,t) = \begin{cases} p, & \text{if } s \overset{p}{\rightsquigarrow} t \\ 1, & \text{if } s = t \text{ and } s \in \mathcal{S}_n \\ 0, & \text{otherwise.} \end{cases}$$

For a set $D \subset \mathcal{S}$ we define $\mu(s, D) = \sum_{t \in D} \mu(s,t)$.

In the bisimulation, we want to compare the reduced density matrices of qubits in output actions, therefore we extend the concepts of density matrix and reduced density matrix to configurations.

Definition 3.2 (Density Matrix of Configurations)**.** Let $\sigma = [\widetilde{pq} \mapsto |\psi\rangle]$ and $s = (\sigma; \omega; P)$. Then

- $\rho(\sigma) = |\psi\rangle\langle\psi|$,
- $\rho^{\widetilde{q}}(\sigma) = \mathrm{tr}_{\widetilde{p}}(|\psi\rangle\langle\psi|)$,
- $\rho(s) = \rho(\sigma)$, and
- $\rho^{\widetilde{q}}(s) = \rho^{\widetilde{q}}(\sigma)$.

Before defining probabilistic branching bisimulation, we introduce some notation that will be used in the remainder of this thesis. Let $\xrightarrow{\tau}{}^{+}$ denote zero or one τ transitions, let $\Longrightarrow$ denote zero or more τ transitions, and let $\overset{\alpha}{\Longrightarrow}$ be equivalent to $\Longrightarrow\xrightarrow{\alpha}\Longrightarrow$.

The following definition is based on the standard definition of branching bisimulation [van Glabbeek and Weijland 1996] with additional conditions for probabilistic configurations, using the function μ, and for matching quantum information. We require the relation $\mathcal{R}$ to be an equivalence relation (instead of a symmetric relation that is normally sufficient) in order to define the equivalence classes $D \in \mathcal{S}/\mathcal{R}$.

Definition 3.3 (Probabilistic Branching Bisimulation)**.** Let s, t be configurations. An equivalence relation $\mathcal{R}$ is a *probabilistic branching bisimulation* on configurations if whenever $(s,t) \in \mathcal{R}$ the following conditions are satisfied.

I. If $s \in \mathcal{S}_n$ and $s \xrightarrow{\tau} s'$ then there exists t', t'' such that $t \Longrightarrow t' \xrightarrow{\tau}{}^{+} t''$ where

   a) $(s, t') \in \mathcal{R}$, and

   b) $(s', t'') \in \mathcal{R}$.

II. If $s \xrightarrow{c![\widetilde{v},\widetilde{q}]} s'$ then $\exists t', t''$ such that $t \Longrightarrow t' \xrightarrow{c![\widetilde{v},\widetilde{r}]} t''$ where

a) $(s, t') \in \mathcal{R}$,

b) $(s', t'') \in \mathcal{R}$,

c) $\rho^{\widetilde{q}}(s') = \rho^{\widetilde{r}}(t'')$.

III. If $s \xrightarrow{c?[\widetilde{v},\widetilde{q}]} s'$ then $\exists t', t''$ such that $t \Longrightarrow t' \xrightarrow{c?[\widetilde{v},\widetilde{r}]} t''$ where

a) $(s, t') \in \mathcal{R}$,

b) $(s', t'') \in \mathcal{R}$,

c) $|\widetilde{v}| = |\widetilde{u}|$, and

d) $\rho^{\widetilde{q}}(s') = \rho^{\widetilde{r}}(t'')$.

IV. If $s \in \mathcal{S}_p$ then $\mu(s, D) = \mu(t, D)$ for all classes $D \in \mathcal{S}/\mathcal{R}$.

Naturally this leads on to the following definition of bisimilarity.

Definition 3.4 (Probabilistic Branching Bisimilarity)**.** Let s and t be configurations. Then s and t are *probabilistic branching bisimilar*, denoted $s \leftrightarrows t$ if and only if there exists a probabilistic branching bisimulation $\mathcal{R}$ such that $(s, t) \in \mathcal{R}$.

Our aim is to define an equivalence for processes, hence we now define bisimilarity for processes based on bisimilarity for configurations. In particular, equivalence for processes should be independent of the quantum state because, unlike a configuration, a process has no quantum state associated with it. The following definition identifies processes that produce bisimilar executions, given any initial quantum state.

Definition 3.5 (Probabilistic Branching Bisimilarity of Processes)**.** Let P and Q be processes. P and Q are *probabilistic branching bisimilar*, denoted $P \leftrightarrows Q$, if and only if for all σ, $(\sigma; \emptyset; P) \leftrightarrows (\sigma; \emptyset; Q)$.

Example 3.1. Let P and Q be processes defined by

$$P = c?[x].\{x *= \mathsf{Z}\}.\{x *= \mathsf{X}\}.d![x].\mathbf{0} \qquad Q = c?[x].\{x *= i\mathsf{Y}\}.d![x].\mathbf{0}$$

Then $P \leftrightarrows Q$ since the identity $\mathsf{ZX} = i\mathsf{Y}$ ensures the state of x upon output will be the same in each process.

We now prove that probabilistic branching bisimilarity of processes is an equivalence relation. This doesn't follow directly from the definition, however it is an important result if we are to use the relation for equational reasoning. A similar proof of transitivity is given by Lalire [2006].

Lemma 3.16. *If $\mathcal{R}$ is a probabilistic branching bisimulation and $s\mathcal{R}t$, and $s \Longrightarrow s'$ then there exists t' such that $t \Longrightarrow t'$ and $(s', t') \in \mathcal{R}$.*

Proof. If $s \Longrightarrow s'$ then there exists a sequence of configurations $s_1, \ldots, s_n$ such that $s \xrightarrow{\tau} s_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} s_n = s'$. The proof is by induction on n.

The base case is $n = 1$. Thus if $s \xrightarrow{\tau} s_1$ then there exist configurations t', t'' such that $t \Longrightarrow t' \xrightarrow{\tau}{}^{+} t''$ where $(s, t') \in \mathcal{R}$ and $(s_1, t'') \in \mathcal{R}$. Equivalently $t \Longrightarrow t''$ as required.

For the inductive step, assume the Lemma holds for n, i.e. if $s \Longrightarrow s_n$ there exists a configuration t' such that $t \Longrightarrow t'$ and $(s_n, t') \in \mathcal{R}$. If $s_n \xrightarrow{\tau} s_{n+1}$ then there exist configurations t'', t''' such that $t' \Longrightarrow t'' \xrightarrow{\tau}{}^{+} t'''$ where $(s_n, t'') \in \mathcal{R}$ and $(s_{n+1}, t''') \in \mathcal{R}$. □

Lemma 3.17. *Probabilistic branching bisimilarity is an equivalence relation.*

Proof. We show that probabilistic branching bisimilarity is reflexive, symmetric, and transitive. The result follows from these properties.

Reflexivity: Let $\mathcal{R}_I$ be the identity relation ($(s, t) \in \mathcal{R}_I$ if and only if $s = t$). It is clear that $\mathcal{R}_I$ is a probabilistic branching bisimulation, hence $\leftrightarrows$ is reflexive.

Symmetry: This follows directly from the symmetry and existence property of the corresponding bisimulation relations. Let P and Q be two processes such that $P \leftrightarrows Q$. By the definition of $\leftrightarrows$ there exists a bisimulation $\mathcal{R}_S$ such that for all $\forall\sigma.((\sigma; \emptyset; P)\mathcal{R}_S(\sigma; \emptyset; Q))$. Since $\mathcal{R}_S$ is symmetric we have that $\forall\sigma.((\sigma; \emptyset; Q)\mathcal{R}_S(\sigma; \emptyset; P))$, hence $Q \leftrightarrows P$.

Transitivity: This does not follow directly from the transitivity of bisimulation. We require the composition of distinct bisimulations to be a bisimulation. Let P, Q and R be processes such that $P \leftrightarrows Q$ and $Q \leftrightarrows R$. We now show that $P \leftrightarrows R$.

From the definition of bisimilarity we know that there exist bisimulations $\mathcal{R}_1$ and $\mathcal{R}_2$ such that for all quantum states σ, $(\sigma; \emptyset; P)\mathcal{R}_1(\sigma; \emptyset; Q)$ and $(\sigma; \emptyset; Q)\mathcal{R}_2(\sigma; \emptyset; R)$. Let $\mathcal{R}_C$ denote the composition of relations $\mathcal{R}_1 \circ \mathcal{R}_2$.

Then let $\mathcal{R}_T$ denote the symmetric and transitive closure of $\mathcal{R}_C$, thus

$$s\mathcal{R}_T u \Rightarrow \exists t_1, \ldots, t_n \mid s = t_0\mathcal{R}_1 t_1 \mathcal{R}_2 t_2 \cdots t_{n-2}\mathcal{R}_1 t_{n-1}\mathcal{R}_2 t_n = u$$

We now show that $\mathcal{R}_T$ is a probabilistic branching bisimulation by induction on n. The base case is $n = 0$ therefore $(s, u) \in \mathcal{R}_T \Rightarrow s = u$.

For the inductive case assume that $\mathcal{R}_T$ is a probabilistic branching bisimulation such that $(s, t_n) \in \mathcal{R}_T$. Therefore there exists $t_1, \ldots, t_{n-1}$ such that $s = t_0\mathcal{R}_1 t_1 \mathcal{R}_2 t_2 \cdots t_{n-2}\mathcal{R}_1 t_{n-1}\mathcal{R}_2 t_n$.

Consider a configuration t_{n+1} such that $(t_n, t_{n+1}) \in \mathcal{R}_1$. The same argument applies respectively if $(t_n, t_{n+1}) \in \mathcal{R}_2$. We consider the four conditions of probabilistic branching bisimulation in turn:

1. If $s \xrightarrow{\tau} s'$ then by $\mathcal{R}_T$ there exist configurations t'_n, t''_n such that $t_n \Longrightarrow t'_n \xrightarrow{\tau}{}^{+} t''_n$ where $(s, t'_n) \in \mathcal{R}_T$ and $(s', t''_n) \in \mathcal{R}_T$. By Lemma 3.16 there exists t'_{n+1}

such that $t_{n+1} \Longrightarrow t'_{n+1}$ and $(t'_n, t'_{n+1}) \in \mathcal{R}_1$. If $t'_n \neq t''_n$ then there exist configurations t''_{n+1}, t'''_{n+1} such that $t'_{n+1} \Longrightarrow t''_{n+1} \xrightarrow{\tau}{}^{+} t'''_{n+1}$ where $(t'_n, t''_{n+1}) \in \mathcal{R}_1$ and $(t''_n, t'''_{n+1}) \in \mathcal{R}_1$. Therefore $s\mathcal{R}_T t'_n \mathcal{R}_1 t''_{n+1}$ and $s'\mathcal{R}_T t''_n \mathcal{R}_1 t'''_{n+1}$. The symmetric property is proved in an identical manner.

2. If $s \xrightarrow{c![\widetilde{u}]} s'$ then by $\mathcal{R}_T$ there exist configurations t'_n, t''_n such that $t_n \Longrightarrow t'_n \xrightarrow{c![\widetilde{v}]} t''_n$ where $(s, t'_n) \in \mathcal{R}_T$ and $(s', t''_n) \in \mathcal{R}_T$. By Lemma 3.16 there exists t'_{n+1} such that $t_{n+1} \Longrightarrow t'_{n+1}$ and $(t'_n, t'_{n+1}) \in \mathcal{R}_1$. Therefore there exist configurations t''_{n+1}, t'''_{n+1} such that $t'_{n+1} \Longrightarrow t''_{n+1} \xrightarrow{c![\widetilde{w}]} t'''_{n+1}$ where $(t'_n, t''_{n+1}) \in \mathcal{R}_1$ and $(t''_n, t'''_{n+1}) \in \mathcal{R}_1$. Therefore $s\mathcal{R}_T t'_n \mathcal{R}_1 t''_{n+1}$, $s'\mathcal{R}_T t''_n \mathcal{R}_1 t'''_{n+1}$ and $\rho^u = \rho^v = \rho^w$. The symmetric property is proved in an identical manner.

3. An identical argument applies to input actions.

4. For the probability function $\mu_{\mathcal{R}}$ we must show that $\mu(s, D) = \mu(u, D)$ for all $D \in \mathcal{S}/\mathcal{R}_T$.

   First we consider the relationship between the equivalence classes of $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_T$, denoted by $\{A_i\}_{i\in I}$, $\{B_j\}_{j\in J}$, and $\{C_k\}_{k\in K}$ respectively. For two states $s, t \in A_i$ for some $i \in I$ we have $s\mathcal{R}_1 t\mathcal{R}_2 t$ using the reflexivity of $\mathcal{R}_2$, thus $s\mathcal{R}_T t$. As a result it must be the case that for each i and k, either $A_i \subseteq C_k$ or $A_i \cap C_k = \emptyset$. Similarly we find that for each j and k it is the case that $B_j \subseteq C_k$ or $B_j \cap C_k = \emptyset$. Furthermore, for each state $s \in C_k$ there exist $i \in I, j \in J$ such that $s \in A_i$ and $s \in B_j$ hence each equivalence class of $\mathcal{R}_T$ is partitioned by some subset $\{A_i\}_{i\in I_k}$ of the equivalence classes of $\mathcal{R}_1$, and separately by a subset $\{B_j\}_{j\in J_k}$ of the equivalence classes of $\mathcal{R}_2$. We can thus say for every C_k:

$$C_k = \bigcup_{i\in I_k} A_i = \bigcup_{j\in J_k} B_j$$

   Therefore, for some s then $\mu(s, C_k) = \sum_{i\in I_k} \mu(s, A_i) = \sum_{j\in J_k} \mu(s, B_j)$. This follows from the definition of μ for both non-deterministic and probabilistic states. Additionally, from the defintion of $\mathcal{R}_T$ we know that for every $s, u \in C_k$ there is a sequence $t_1, \ldots, t_m$ of states in C_k such that $s\mathcal{R}_1 t_1 \mathcal{R}_2 t_2 \mathcal{R}_1 \cdots \mathcal{R}_1 t_m \mathcal{R}_2 u$.

   By induction on m: the base case ($m = 0$) is trivial. Assume that $\mu(s, C_k) = \mu(t_m, C_k)$ and $(t_m, t_{m+1}) \in \mathcal{R}_1$. Then $\mu(t_{m+1}, C_k) = \sum_{i\in I_k} \mu(t_{m+1}, A_i)$. Since $\mathcal{R}_1$ is a bisimulation, for each $i \in I_k$, $\mu(t_{m+1}, A_i) = \mu(t_m, A_i)$, hence it follows that $\mu(t_{m+1}, C_k) = \mu(t_m, C_k) = \mu(s, C_k)$.

We have now shown that the relation $\mathcal{R}_T$ is a bisimulation, therefore we have for all σ, $((\sigma; \emptyset; P), (\sigma; \emptyset; R)) \in \mathcal{R}_T$, hence $P \leftrightarrows R$. This completes the proof. □

$$
\begin{array}{rcl}
\mathit{Teleport} & = & (\mathsf{qbit}\ y, z)(\{z \mathrel{*}\!= \mathsf{H}\}.\{z, y \mathrel{*}\!= \mathsf{CNot}\}.(\nu e\!:\!\widehat{}[\mathsf{Int}, \mathsf{Int}])(\mathit{Alice} \parallel \mathit{Bob})) \\
\mathit{Alice} & = & c?[x\!:\!\mathsf{Qbit}].\{x, z \mathrel{*}\!= \mathsf{CNot}\}.\{x \mathrel{*}\!= \mathsf{H}\}.e![\mathsf{measure}\ z, \mathsf{measure}\ x].\mathbf{0} \\
\mathit{Bob} & = & e?[r\!:\!\mathsf{Int}, s\!:\!\mathsf{Int}].\{y \mathrel{*}\!= X^r\}.\{y \mathrel{*}\!= Z^s\}.d![y].\mathbf{0} \\
\mathit{QChannel} & = & c?[x\!:\!\mathsf{Qbit}].d![x].\mathbf{0}
\end{array}
$$

Figure 3.8. Quantum teleportation modelled in CQP (*Teleport*) and its specification process (*QChannel*).

3.3 Applications

In this section, we demonstrate the use of bisimilarity for verification by applying the probabilistic branching bisimilarity to several processes. Quantum teleportation was used as an example to motivate our approach to modelling communication protocols. We will present a CQP model for teleportation, and formally define a specification process (*QChannel*) that describes its high-level observational behaviour. Verification of teleportation is then achieved by proving that the two processes are bisimilar. We will also consider an alternate teleportation protocol in which the measurements are deferred, and a qubit-swap circuit. By showing each process is bisimilar to *QChannel* and because bisimilarity is an equivalence relation, we demonstrate that these processes are all observationally equivalent to one another.

3.3.1 Quantum Teleportation

The quantum teleportation protocol was described in Section 2.2.1, including a circuit model representation. A CQP model of teleportation is given by the process *Teleport* in Figure 3.8. This model clearly specifies the individual users Alice and Bob, who act in parallel with one another (*Alice* $\parallel$ *Bob*). The quantum state to be teleported is received by Alice on channel c, and at the end of the protocol, Bob sends his qubit on channel d. These inputs and outputs represent the external interactions. The communication between Alice and Bob uses channel e, which is restricted to the two processes and therefore does not present a possibility for external interaction.

Also given in Figure 3.8 is the process *QChannel*, which is the high-level specification process for teleportation. This process acts like a quantum channel; a qubit is received at one end-point (channel c), and sent at the other (channel d). In particular, no operations are performed in between, hence it is obvious that the input and output states are identical.

Lemma 3.18. *QChannel* $\leftrightarrows$ *Teleport*.

Proof. To prove that *QChannel* $\leftrightarrows$ *Teleport*, we define an equivalence relation $\mathcal{R}$ that contains $((\sigma; \emptyset; \mathit{Teleport}), (\sigma; \emptyset; \mathit{QChannel}))$ for all σ, and then show that it is a

bisimulation.

For $P \in \{QChannel, Teleport\}$ and configurations s, define the sets $D_1(\sigma)$, $D_2(\sigma)$ and $D_3(\sigma)$ as follows:

- $s \in D_1(\sigma)$ if and only if $s = (\sigma'; \omega'; P')$ and $(\sigma; \emptyset; P) \Longrightarrow (\sigma'; \omega'; P')$.
- $s \in D_2(\sigma)$ if and only if $s = \boxplus_i\, p_i \bullet (\sigma_i; \omega_i; P_i)$ and $(\sigma; \emptyset; P) \stackrel{c?[p]}{\Longrightarrow} \boxplus_i\, p_i \bullet (\sigma_i; \omega_i; P_i)$.
- $s \in D_3(\sigma)$ if and only if $s = (\sigma'; \omega'; P')$ and $(\sigma; \emptyset; P) \stackrel{c?[p]}{\Longrightarrow}\stackrel{d![q]}{\Longrightarrow} (\sigma'; \omega'; P')$.

Let $\mathcal{R}$ be an equivalence relation such that $(s,t) \in \mathcal{R}$ if there exists i and σ such that $s, t \in D_i(\sigma)$.

Consider a general quantum state $\sigma = [\widetilde{rp} \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]$. We consider the pairs of configurations in each of the equivalence classes $D_1(\sigma)$, $D_2(\sigma)$ and $D_3(\sigma)$. The configurations in $D_1(\sigma)$ are as follows.

$D_1([\widetilde{rp} \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]) :$

$s_{11}:\quad ((\widetilde{rp} \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle)]; \emptyset; Teleport)$

$s_{12}:\quad ((\widetilde{rpq_1q_2} \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle) \otimes |00\rangle)]; q_1, q_2; (\{q_2 *= \mathsf{H}\}.\{q_2, q_1 *= \mathsf{CNot}\}.$
$(\nu e{:}\widehat{}[\mathsf{Int}, \mathsf{Int}])(Alice \parallel Bob)))$

$s_{13}:\quad ((\widetilde{rpq_1q_2} \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle))]; q_1, q_2; \{q_2, q_1 *= \mathsf{CNot}\}.$
$(\nu e{:}\widehat{}[\mathsf{Int}, \mathsf{Int}])(Alice \parallel Bob))$

$s_{14}:\quad ((\widetilde{rpq_1q_2} \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle))]; q_1, q_2; Alice \parallel Bob)$

$s_{15}:\quad ((\widetilde{rp} \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle)]; \emptyset; QChannel)$

For $s \in \{s_{11}, s_{12}, s_{13}\}$ we have $s \stackrel{\tau}{\longrightarrow} s'$ where $s' \in D_1(\sigma)$. For all $t \in D_1(\sigma)$ let $t = t' = t''$ then $t \Longrightarrow t' \stackrel{\tau}{\longrightarrow}^{+} t''$ and $t', t'' \in D_1(\sigma)$.

For $s \in \{s_{14}, s_{15}\}$ we have $s \stackrel{c?[p]}{\longrightarrow} s'$ where $s' \in D_2(\sigma)$ and $\rho^p = \sum_{i,j\in\{0,1\}} \langle\psi_i|\psi_j\rangle |i\rangle\langle j|$. For any $t \in D_1(\sigma)$ there exists t', t'' such that $t \Longrightarrow t' \stackrel{c?[p]}{\longrightarrow} t''$ with $t'' \in D_2(\sigma)$ and $\rho^p = \sum_{i,j\in\{0,1\}} \langle\psi_i|\psi_j\rangle |i\rangle\langle j|$.

Now we consider the configurations in the class $D_2(\sigma)$. These include a probabilistic distribution and configurations that arise through probabilistic branching.

$D_2([\widetilde{rp} \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]) :$

$s_{21}:\quad ((\widetilde{rpq_1q_2} \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle))]; q_1, q_2, p; \{q_2, p *= \mathsf{CNot}\}.$
$\{q_2 *= \mathsf{H}\}.e![\mathsf{measure}\ q_2, \mathsf{measure}\ p].\mathbf{0} \parallel Bob)$

$s_{22}:\quad ([\widetilde{rpq_1q_2} \mapsto \frac{1}{\sqrt{2}}|\psi_0\rangle(|000\rangle + |111\rangle) + \frac{1}{\sqrt{2}}|\psi_1\rangle(|100\rangle + |011\rangle)]; q_1, q_2, p;$
$\{q_2 *= \mathsf{H}\}.e![\mathsf{measure}\ q_2, \mathsf{measure}\ p].\mathbf{0} \parallel Bob)$

$$
\begin{aligned}
s_{23}: \quad & ([\widetilde{r}pq_1q_2 \mapsto \tfrac{1}{2}|\psi_0\rangle(|000\rangle + |001\rangle + |110\rangle - |111\rangle) + \\
& + \tfrac{1}{2}|\psi_1\rangle(|100\rangle + |101\rangle + |010\rangle - |011\rangle)]; q_1, q_2, p; \\
& e![\mathsf{measure}\ q_2, \mathsf{measure}\ p].\mathbf{0} \parallel Bob) \\
s_{24}: \quad & \tfrac{1}{4} \bullet ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|000\rangle + |\psi_1\rangle|010\rangle]; q_1, q_2, p; e![0,0].\mathbf{0} \parallel Bob) \\
& \boxplus\ \tfrac{1}{4} \bullet ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|110\rangle + |\psi_1\rangle|100\rangle]; q_1, q_2, p; e![0,1].\mathbf{0} \parallel Bob) \\
& \boxplus\ \tfrac{1}{4} \bullet ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|001\rangle - |\psi_1\rangle|011\rangle]; q_1, q_2, p; e![1,0].\mathbf{0} \parallel Bob) \\
& \boxplus\ \tfrac{1}{4} \bullet ([\widetilde{r}pq_1q_2 \mapsto -|\psi_0\rangle|111\rangle + |\psi_1\rangle|101\rangle]; q_1, q_2, p; e![1,1].\mathbf{0} \parallel Bob) \\
s_{25}: \quad & ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|000\rangle + |\psi_1\rangle|010\rangle]; q_1, q_2, p; e![0,0].\mathbf{0} \parallel Bob) \\
s_{26}: \quad & ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|110\rangle + |\psi_1\rangle|100\rangle]; q_1, q_2, p; e![0,1].\mathbf{0} \parallel Bob) \\
s_{27}: \quad & ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|001\rangle - |\psi_1\rangle|011\rangle]; q_1, q_2, p; e![1,0].\mathbf{0} \parallel Bob) \\
s_{28}: \quad & ([\widetilde{r}pq_1q_2 \mapsto -|\psi_0\rangle|111\rangle + |\psi_1\rangle|101\rangle]; q_1, q_2, p; e![1,1].\mathbf{0} \parallel Bob) \\
s_{29}: \quad & ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|000\rangle + |\psi_1\rangle|010\rangle]; q_1, q_2, p; \mathbf{0} \parallel \{q_1 *= Z^0X^0\}.d![q_1].\mathbf{0}) \\
s_{210}: \quad & ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|110\rangle + |\psi_1\rangle|100\rangle]; q_1, q_2, p; \mathbf{0} \parallel \{q_1 *= Z^0X^1\}.d![q_1].\mathbf{0}) \\
s_{211}: \quad & ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|001\rangle - |\psi_1\rangle|011\rangle]; q_1, q_2, p; \mathbf{0} \parallel \{q_1 *= Z^1X^0\}.d![q_1].\mathbf{0}) \\
s_{212}: \quad & ([\widetilde{r}pq_1q_2 \mapsto -|\psi_0\rangle|111\rangle + |\psi_1\rangle|101\rangle]; q_1, q_2, p; \mathbf{0} \parallel \{q_1 *= Z^1X^1\}.d![q_1].\mathbf{0}) \\
s_{213}: \quad & ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|000\rangle + |\psi_1\rangle|010\rangle]; q_1, q_2, p; d![q_1].\mathbf{0}) \\
s_{214}: \quad & ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|001\rangle + |\psi_1\rangle|011\rangle]; q_1, q_2, p; d![q_1].\mathbf{0}) \\
s_{215}: \quad & ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|100\rangle + |\psi_1\rangle|110\rangle]; q_1, q_2, p; d![q_1].\mathbf{0}) \\
s_{216}: \quad & ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|101\rangle + |\psi_1\rangle|111\rangle]; q_1, q_2, p; d![q_1].\mathbf{0}) \\
s_{217}: \quad & ([\widetilde{r}p \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]; \emptyset; d![p].\mathbf{0})
\end{aligned}
$$

For $s \in \{s_{21}, s_{22}, s_{23}, s_{25}, \cdots, s_{212}\}$ we have $s \xrightarrow{\tau} s'$ where $s' \in D_2(\sigma)$. For any $t \in D_2(\sigma)$ let $t = t' = t''$ then $t \Longrightarrow t' \xrightarrow{\tau}{}^{+} t''$ and $t', t'' \in D_2(\sigma)$.

For $s = s_{24}$ we have $s \overset{\frac{1}{4}}{\rightsquigarrow} s_i$ for $s_i \in \{s_{25}, \cdots, s_{28}\}$ therefore $\mu(s, D_2(\sigma)) = 1$. For all $t \in D_2(\sigma) \setminus s_{24}$ we have $\mu(t, D_2(\sigma)) = 1$ since $t \in \mathcal{S}_n$ and $t \in D_2(\sigma)$.

For $s \in \{s_{213}, \cdots, s_{216}\}$ we have $s \xrightarrow{d![q_1]} s'$ where $\rho^{q_1} = \sum_{i,j\in\{0,1\}} \langle\psi_i|\psi_j\rangle |i\rangle\langle j|$ and $s' \in D_3(\sigma)$. For $t = s_{217}$ we have $t \xrightarrow{d![p]} t'$ where $t' \in D_3(\sigma)$ and $\rho^p = \sum_{i,j\in\{0,1\}} \langle\psi_i|\psi_j\rangle |i\rangle\langle j|$. For all other $u \in D_2(\sigma)$ there exists u', u'' such that $u \Longrightarrow u' \xrightarrow{d![q_1]} u''$ and $u' \in D_2(\sigma)$ and $u'' \in D_3(\sigma)$ and $\rho^{q_1} = \sum_{i,j\in\{0,1\}} \langle\psi_i|\psi_j\rangle |i\rangle\langle j|$.

For $s \in D_3(\sigma)$ there are no possible transitions. These configurations are as

$$Teleport_D = (\mathsf{qbit}\ y, z)(\{z *\!\!= \mathsf{H}\}.\{z, y *\!\!= \mathsf{CNot}\}.Alice)$$
$$Alice = c?[x].\{x, y *\!\!= \mathsf{CNot}\}.\{x *\!\!= \mathsf{H}\}.\{y, z *\!\!= \mathsf{CX}\}.\{x, z *\!\!= \mathsf{CZ}\}.d![z].\mathbf{0}$$

Figure 3.9. Quantum teleportation with deferred measurement.

follows.

$$\begin{aligned}
&D_3([\widetilde{r}p \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]) : \\
&\quad s_{31} : \quad ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|000\rangle + |\psi_1\rangle|010\rangle]; q_1, q_2; \mathbf{0}) \\
&\quad s_{32} : \quad ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|001\rangle + |\psi_1\rangle|011\rangle]; q_1, q_2; \mathbf{0}) \\
&\quad s_{33} : \quad ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|100\rangle + |\psi_1\rangle|110\rangle]; q_1, q_2; \mathbf{0}) \\
&\quad s_{34} : \quad ([\widetilde{r}pq_1q_2 \mapsto |\psi_0\rangle|101\rangle + |\psi_1\rangle|111\rangle]; q_1, q_2; \mathbf{0}) \\
&\quad s_{35} : \quad ([\widetilde{r}p \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]; \emptyset; \mathbf{0})
\end{aligned}$$

We therefore have $((\sigma, \emptyset; Teleport), (\sigma; \emptyset; QChannel)) \in \mathcal{R}$. We can express all quantum states in the form of σ, hence this condition holds for all σ. □

3.3.2 Quantum Teleportation with Deferred Measurement

The *principle of deferred measurement* states that measurements can be moved from an intermediate stage of a quantum circuit to the end of the circuit. If the result of the measurement is used at any stage within the circuit, then the classically controlled operation may be replaced by a controlled quantum gate. Applying this principle to the quantum teleportation protocol results in a similar protocol, defined in Figure 3.9.

In this modified version the protocol has no logical split between two parties – it is not really teleportation as such. Instead there is only one user, Alice, who owns all the qubits. Using Lemma 3.18 and the fact that bisimilarity is transitive, we are able to show that this protocol is bisimilar to the original teleportation process *Teleport*.

Lemma 3.19. *$Teleport_D \leftrightarrows QChannel$ and $Teleport_D \leftrightarrows Teleport$.*

Proof. Following a similar course as the proof of Lemma 3.18, we consider an arbitrary quantum state $\sigma = [\widetilde{q}p \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]$. We define an equivalence relation, $\mathcal{R}$, by

the following parameterised equivalence classes.

$D_1(\sigma)$:

$([\widetilde{r}p \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]; \emptyset; \mathit{QChannel})$

$([\widetilde{r}p \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]; \emptyset; \mathit{Teleport}_D)$

$([\widetilde{r}pq_1q_2 \mapsto (|\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle) \otimes |00\rangle]; q_1, q_2;$
$\{q_2 *= \mathsf{H}\}.\{q_2, q_1 *= \mathsf{CNot}\}.\mathit{Alice}\{q_1q_2/yz\})$

$([\widetilde{r}pq_1q_2 \mapsto (|\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)]; q_1, q_2;$
$\{q_2, q_1 *= \mathsf{CNot}\}.\mathit{Alice}\{q_1q_2/yz\})$

$([\widetilde{r}pq_1q_2 \mapsto (|\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; q_1, q_2; \mathit{Alice}\{q_1q_2/yz\})$

$D_2(\sigma)$:

$([\widetilde{r}p \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]; p; d![p].\mathbf{0})$

$([\widetilde{r}pq_1q_2 \mapsto (|\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; p, q_1, q_2;$
$\{p, q_1 *= \mathsf{CNot}\}.\{x *= \mathsf{H}\}.\{q_1, q_2 *= \mathsf{CX}\}.\{p, q_2 *= \mathsf{CZ}\}.d![q_2].\mathbf{0})$

$([\widetilde{r}pq_1q_2 \mapsto \frac{1}{\sqrt{2}}(|\psi_0\rangle|000\rangle + |\psi_0\rangle|011\rangle + |\psi_1\rangle|110\rangle + |\psi_1\rangle|101\rangle)]; p, q_1, q_2;$
$\{p *= \mathsf{H}\}.\{q_1, q_2 *= \mathsf{CX}\}.\{p, q_2 *= \mathsf{CZ}\}.d![q_2].\mathbf{0})$

$([\widetilde{r}pq_1q_2 \mapsto \frac{1}{2}(|\psi_0\rangle|000\rangle + |\psi_0\rangle|100\rangle + |\psi_0\rangle|011\rangle + |\psi_0\rangle|111\rangle + |\psi_1\rangle|010\rangle - |\psi_1\rangle|110\rangle$
$+ |\psi_1\rangle|001\rangle - |\psi_1\rangle|101\rangle)]; p, q_1, q_2; \{q_1, q_2 *= \mathsf{CX}\}.\{p, q_2 *= \mathsf{CZ}\}.d![q_2].\mathbf{0})$

$([\widetilde{r}pq_1q_2 \mapsto \frac{1}{2}(|\psi_0\rangle|000\rangle + |\psi_0\rangle|100\rangle + |\psi_0\rangle|010\rangle + |\psi_0\rangle|110\rangle + |\psi_1\rangle|011\rangle - |\psi_1\rangle|111\rangle$
$+ |\psi_1\rangle|001\rangle - |\psi_1\rangle|101\rangle)]; p, q_1, q_2; \{p, q_2 *= \mathsf{CZ}\}.d![q_2].\mathbf{0})$

$([\widetilde{r}pq_1q_2 \mapsto \frac{1}{2}(|00\rangle + |10\rangle + |01\rangle + |11\rangle) \otimes (|\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle)]; p, q_1, q_2; d![q_2].\mathbf{0})$

$D_3(\sigma)$:

$([\widetilde{r}p \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]; \emptyset; \mathbf{0})$

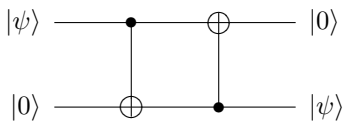
$([\widetilde{r}pq_1q_2 \mapsto \frac{1}{2}(|00\rangle + |10\rangle + |01\rangle + |11\rangle) \otimes (|\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle)]; q_1, q_2; \mathbf{0})$

By reasoning similar to Lemma 3.18 the relation $\mathcal{R}$ is a probabilistic branching bisimulation, and hence $\mathit{QChannel} \leftrightarrows \mathit{Teleport}_D$. Furthermore, because bisimilarity is transitive, we have $\mathit{Teleport} \leftrightarrows \mathit{Teleport}_D$. □

It is worth noting that the measurement operation has been omitted, instead making use of the *principle of implicit measurement*. It is easily seen that including explicit measurement of the qubits x and y would not affect this equivalence. Both the principles of deferred and implicit measurement will be revisited in Chapter 5, where we formulate them as axioms of equivalence.

3.3.3 Qubit-Swap Circuit

The final example we consider is a state-swapping circuit.



This circuit has one input qubit in state $|\psi\rangle$ and a second qubit starting in the basis state $|0\rangle$. By performing two CNot operations, the state of the first qubit is transferred to the second qubit, leaving the first qubit in state $|0\rangle$. This circuit is a simplification of the state-swapping circuit that swaps two arbitrary quantum states by using a third CNot gate with the first qubit as the control. In this case, since the first qubit will be in state $|0\rangle$ at this point, a third CNot operation would have no effect.

Quantum circuits can be manipulated using a number of identity relations on sub-components. For example the control and target of a CNot operation can be switched as shown in Figure 3.10. Note that the CNot gate is the same as a controlled-X gate.

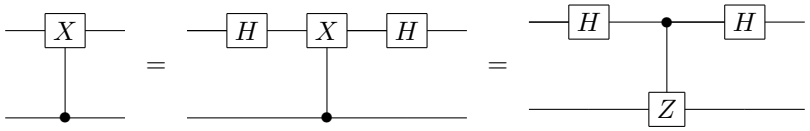


Figure 3.10. Circuit identities for switching the control and target qubits.

It has been shown by Mermin [2001], that by manipulating the state-swapping circuit using circuit identities, it is possible to obtain a quantum teleportation circuit. The swap circuit can be represented by the CQP process *Swap*, where the state of the first qubit x is to be transferred to the second qubit y.

$$Swap = (\mathsf{qbit}\ y)c?[x].\{x, y *= \mathsf{CNOT}\}.\{y, x *= \mathsf{CNOT}\}.d![y].\mathbf{0}$$

This circuit equivalence can be expressed and proved with respect to bisimilarity in CQP.

Lemma 3.20. *Swap* $\leftrightarrows$ *Teleport*.

Proof. We first prove that *Swap* $\leftrightarrows$ *QChannel*. Then by the transitivity of bisimilarity, we obtain *Swap* $\leftrightarrows$ *Teleport*.

Consider an arbitrary quantum state $\sigma = [\widetilde{qp} \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]$. Then define an

equivalence relation $\mathcal{R}$ by the following equivalence classes.

$D_1(\sigma)$:

$s_{11} = ([\widetilde{qp} \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]; \emptyset; \mathit{QChannel})$

$s_{12} = ([\widetilde{qp} \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]; \emptyset; \mathit{Swap})$

$s_{13} = ([\widetilde{qpr} \mapsto |\psi_0\rangle|00\rangle + |\psi_1\rangle|10\rangle]; r; c?[x].\{x, r *= \mathsf{CNOT}\}.\{r, x *= \mathsf{CNOT}\}.d![r].\mathbf{0})$

$D_2(\sigma)$:

$s_{21} = ([\widetilde{qp} \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]; p; d![p].\mathbf{0})$

$s_{22} = ([\widetilde{qpr} \mapsto |\psi_0\rangle|00\rangle + |\psi_1\rangle|10\rangle]; p, r; \{p, r *= \mathsf{CNOT}\}.\{r, p *= \mathsf{CNOT}\}.d![r].\mathbf{0})$

$s_{23} = ([\widetilde{qpr} \mapsto |\psi_0\rangle|00\rangle + |\psi_1\rangle|11\rangle]; p, r; \{r, p *= \mathsf{CNOT}\}.d![r].\mathbf{0})$

$s_{24} = ([\widetilde{qpr} \mapsto |\psi_0\rangle|00\rangle + |\psi_1\rangle|01\rangle]; p, r; d![r].\mathbf{0})$

$D_3(\sigma)$:

$s_{31} = ([\widetilde{qp} \mapsto |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle]; \emptyset; \mathbf{0})$

$s_{32} = ([\widetilde{qpr} \mapsto |\psi_0\rangle|00\rangle + |\psi_1\rangle|01\rangle]; p; \mathbf{0})$

For each configuration in $s_i \in D_1(\sigma)$ there are transitions $s_i \Longrightarrow s_i' \xrightarrow{c?[x]} s_i''$ where $s_i' \in D_1(\sigma)$ and $s_i' \in D_2(\sigma)$ and $\rho^p = \sum_{i,j\in\{0,1\}} \langle\psi_i|\psi_j\rangle |i\rangle\langle j|$. These transitions can therefore match the input transitions of s_{11} and s_{13}. The internal transition $s_{12} \xrightarrow{\tau} s_{13}$ can be matched by zero transitions, since $s_{13} \in D_1(\sigma)$.

The internal transitions $s_{22} \xrightarrow{\tau} s_{23}$ and $s_{23} \xrightarrow{\tau} s_{24}$ can both be matched by zero transitions, since $s_{22}, s_{23}, s_{24} \in D_2(\sigma)$.

For each configuration s_{22}, s_{23}, s_{24} there are transitions $s_i \Longrightarrow s_{24} \xrightarrow{d![y]} s_{32}$ where $s_{24} \in D_2(\sigma)$ and $s_{32} \in D_3(\sigma)$ and $\rho^r = \sum_{i,j\in\{0,1\}} \langle\psi_i|\psi_j\rangle |i\rangle\langle j|$. Similarly there is a transition $s_{21} \xrightarrow{d![x]} s_{31}$ where $\rho^p = \sum_{i,j\in\{0,1\}} \langle\psi_i|\psi_j\rangle |i\rangle\langle j|$. These transitions are able to match the two output transitions of s_{21} and s_{24}. Thus, the possible transitions corresponding to each of the four configurations in $D_2(\sigma)$ can be matched by all other configurations in $D_2(\sigma)$.

Finally s_{31} and s_{32} have no possible transitions. We therefore conclude that the equivalence relation $\mathcal{R}$ is a probabilistic branching bisimulation. Furthermore $\mathit{Swap} \leftrightarrows \mathit{QChannel}$, since for any σ, $(\sigma; \emptyset; \mathit{Swap})\mathcal{R}(\sigma; \emptyset; \mathit{QChannel})$. Due to the transitivity of bisimilarity, this gives $\mathit{Swap} \leftrightarrows \mathit{Teleport}$. □

3.4 Congruence Properties

We have introduced probabilistic branching bisimilarity for CQP, and in the previous section we applied it to several protocols. In this section, we consider whether this bisimilarity is a congruence relation. A congruence relation is particularly useful,

because it ensures that equivalence is maintained in any context. For example, if the two teleportation protocols considered in the previous section are congruent, then they are considered interchangeable components in a composite system.

We begin by formally defining a context. A context is similar to a process, however it contains a *hole* indicating the position in which a process can be placed. A context describes an environment and, through the substitution of a process into the hole, results in the complete representation of that process in the given environment. A context is defined to allow an environment access to the full power of quantum mechanics, including the initialisation of new qubits into the state. This is significant when considering the effects of a parallel context on bisimilar processes, since we are not placing any limits on the capabilities of the environment.

Definition 3.6 (Context)**.** A *context* is a process expression in which an occurrence of **0** is replaced by a *hole*, [ ]. Formally, a context is given by the grammar

$$C ::= [\,] \mid (C \parallel P) \mid \alpha.C + P \mid \alpha.C \mid (\nu x{:}\widehat{\ }[T])C \mid (\mathsf{qbit}\ x)C$$

where $\alpha \in \{e?[\tilde{x} : \tilde{T}], e![\tilde{e}], \{e\}\}$. Given a process P, $C[P]$ denotes the process resulting from filling the hole in C by P.

Due to the possibility of a context capturing the free names of a process, we extend the notion of bisimilarity to *full bisimilarity* before considering preservation by prefixes. In general, bisimilarity is not preserved by prefixes that capture free names; these consist of input and qubit declaration. Full bisimilarity is an extension of bisimilarity that requires the relation to hold for all substitutions.

Definition 3.7 (Full probabilistic branching bisimilarity)**.** Processes P and Q are *full probabilistic branching bisimilar*, denoted $P \leftrightarrows^c Q$, if for any substitution $\kappa = \{\widetilde{u}, \widetilde{q}/\widetilde{x}\}$ and for any quantum state σ, $(\sigma; \widetilde{q}; P\kappa) \leftrightarrows (\sigma; \widetilde{q}; Q\kappa)$.

To prove that full probabilistic branching bisimilarity is preserved by prefixes, we define a *prefix context*. We will consider preservation by parallel composition in Section 3.4.1.

Definition 3.8 (Prefix context)**.** A prefix context is a context without parallel composition, specified by the grammar

$$C ::= [\,] \mid \alpha.C + P \mid \alpha.C \mid (\nu x{:}\widehat{\ }[T])C \mid (\mathsf{qbit}\ x)C$$

where $\alpha \in \{e?[\tilde{x} : \tilde{T}], e![\tilde{e}], \{e\}\}$.

The following two lemmas will be used in the proof of Lemma 3.23 (preservation by prefix contexts).

Lemma 3.21 (Weakening for qubit list)**.** *If* $(\sigma;\omega_1;P) \xrightarrow{\alpha} \boxplus_i\ p_i \bullet (\sigma_i;\omega_1';P_i)$ *and* $\omega_1 \cap \omega_2 = \omega_1' \cap \omega_2 = \emptyset$ *then* $(\sigma;\omega_1,\omega_2;P) \xrightarrow{\alpha} \boxplus_i\ p_i \bullet (\sigma_i;\omega_1',\omega_2;P_i)$.

Proof. By induction on the derivation of $(\sigma;\omega_1;P) \xrightarrow{\alpha} \boxplus_i\ p_i \bullet (\sigma_i;\omega_1';P_i)$. The interesting cases are L-Out, L-In, L-Com and L-Qbit; the rest are straightforward.

L-Out: We have $(\sigma;\omega_1,\widetilde{q};P) \xrightarrow{c![\widetilde{q}]} (\sigma;\omega_1;P')$. Because $\omega_1,\widetilde{q} \cap \omega_2 = \emptyset$ we obtain $(\sigma;\omega_1,\widetilde{q},\omega_2;P) \xrightarrow{c![\widetilde{q}]} (\sigma;\omega_1,\omega_2;P')$.

L-In: We have $(\sigma;\omega_1;P) \xrightarrow{c?[\widetilde{q}]} (\sigma;\omega_1,\widetilde{q};P')$. Because $\omega_1,\widetilde{q} \cap \omega_2 = \emptyset$ we obtain $(\sigma;\omega_1,\omega_2;P) \xrightarrow{c?[\widetilde{q}]} (\sigma;\omega_1,\widetilde{q},\omega_2;P')$.

L-Com: We have

$$\frac{(\sigma;\omega_1,\widetilde{q};P) \xrightarrow{c![\widetilde{v}]} (\sigma;\omega_1;P') \quad (\sigma;\omega_1;Q) \xrightarrow{c?[\widetilde{v}]} (\sigma;\omega_1,\widetilde{q};Q')}{(\sigma;\omega_1,\widetilde{q};P \parallel Q) \xrightarrow{\tau} (\sigma;\omega_1,\widetilde{q};P' \parallel Q')}$$

Applying the inductive hypothesis gives transitions $(\sigma;\omega_1,\widetilde{q},\omega_2;P) \xrightarrow{c![\widetilde{v}]} (\sigma;\omega_1,\omega_2;P')$ and $(\sigma;\omega_1,\omega_2;Q) \xrightarrow{c?[\widetilde{v}]} (\sigma;\omega_1,\widetilde{q},\omega_2;Q')$. Therefore (by L-Com) we obtain the transition $(\sigma;\omega_1,\widetilde{q},\omega_2;P \parallel Q) \xrightarrow{\tau} (\sigma;\omega_1,\widetilde{q},\omega_2;P' \parallel Q')$.

L-Qbit: We have $(\sigma;\omega_1;P) \xrightarrow{\tau} (\sigma';\omega_1,q;P')$. Because $\omega_1,q \cap \omega_2 = \emptyset$ we obtain $(\sigma;\omega_1,\omega_2;P) \xrightarrow{\tau} (\sigma';\omega_1,q,\omega_2;P')$ such that q is fresh. □

Lemma 3.22 (Strengthening for qubit list)**.** *If* $\Gamma;\omega_1 \vdash P$ *and* $(\sigma;\omega_1,\omega_2;P) \xrightarrow{\alpha} (\sigma';\omega_1',\omega_2;P')$ *then* $(\sigma;\omega_1;P) \xrightarrow{\alpha} (\sigma';\omega_1';P')$

Proof. By induction on the derivation of $(\sigma;\omega_1,\omega_2;P) \xrightarrow{\tau} (\sigma';\omega_1',\omega_2;P')$. The interesting cases are L-Out, L-In, L-Com and L-Qbit; the rest are straightforward.

L-Out: We have $(\sigma;\omega_1,\widetilde{q},\omega_2;P) \xrightarrow{c![\widetilde{q}]} (\sigma;\omega_1,\omega_2;P')$ and $\Gamma;\omega_1,\widetilde{q} \vdash P$. Therefore $(\sigma;\omega_1,\widetilde{q};P) \xrightarrow{c![\widetilde{q}]} (\sigma;\omega_1;P')$.

L-In: We have $(\sigma;\omega_1,\omega_2;P) \xrightarrow{c?[\widetilde{q}]} (\sigma;\omega_1,\widetilde{q},\omega_2;P')$ and $\Gamma;\omega_1 \vdash P$. Therefore $(\sigma;\omega_1;P) \xrightarrow{c?[\widetilde{q}]} (\sigma;\omega_1,\widetilde{q};P')$.

L-Com: We have

$$\frac{(\sigma;\omega_1,\widetilde{q},\omega_2;P) \xrightarrow{c![\widetilde{v}]} (\sigma;\omega_1,\omega_2;P') \quad (\sigma;\omega_1,\omega_2;Q) \xrightarrow{c?[\widetilde{v}]} (\sigma;\omega_1,\widetilde{q},\omega_2;Q')}{(\sigma;\omega_1,\widetilde{q},\omega_2;P \parallel Q) \xrightarrow{\tau} (\sigma;\omega_1,\widetilde{q},\omega_2;P' \parallel Q')}$$

The typing derivation of $\Gamma;\omega_1,\widetilde{q} \vdash P \parallel Q$ has hypotheses $\Gamma_1;\omega_p,\widetilde{q} \vdash P$ and $\Gamma_2;\omega_q \vdash Q$ where $\omega_1 = \omega_p \cup \omega_q$ and $\omega_p,\widetilde{q} \cap \omega_q = \emptyset$. The inductive hypothesis gives $(\sigma;\omega_p,\widetilde{q};P) \xrightarrow{c![\widetilde{v}]} (\sigma;\omega_p;P')$ and $(\sigma;\omega_q;Q) \xrightarrow{c?[\widetilde{v}]} (\sigma;\omega_q,\widetilde{q};Q')$. From Lemma 3.21 we have $(\sigma;\omega_1,\widetilde{q};P) \xrightarrow{c![\widetilde{v}]} (\sigma;\omega_1;P')$ and $(\sigma;\omega_1;Q) \xrightarrow{c?[\widetilde{v}]} (\sigma;\omega_1,\widetilde{q};Q')$. Therefore (by L-Com) this gives the transition $(\sigma;\omega_1,\widetilde{q};P \parallel Q) \xrightarrow{\tau} (\sigma;\omega_1,\widetilde{q};P' \parallel Q')$.

L-Qbit: We have $(\sigma;\omega_1,\omega_2;P) \xrightarrow{\tau} (\sigma';\omega_1,\omega_2,q;P')$. Because q is fresh we have $q \notin \omega_2$. Therefore $(\sigma;\omega_1;P) \xrightarrow{\tau} (\sigma';\omega_1,q;P')$ such that q is fresh. □

Lemma 3.23 (Preservation by prefix contexts)**.** *If $P \leftrightarrows^c Q$ and then for any prefix context C, $C[P] \leftrightarrows^c C[Q]$, provided that $\Gamma \vdash C[P]$ and $\Gamma \vdash C[Q]$.*

Proof. By induction on the structure of C. The inductive hypothesis gives a bisimulation $\mathcal{R}$, such that for all σ and $\kappa = \{\widetilde{v}, \widetilde{q}/\widetilde{x}\}$ we have $((\sigma; \widetilde{q}; C[P]\kappa), (\sigma; \widetilde{q}; C[Q]\kappa)) \in \mathcal{R}$.

Input: Let $C = c?[\widetilde{x}_1].C'$ and $\kappa_1 = \{\widetilde{v}_1, \widetilde{q}_1/\widetilde{x}_1\}$ and $\kappa_2 = \{\widetilde{v}_2, \widetilde{q}_2/\widetilde{x}_2\}$ and $\mathcal{R}' = ((\sigma; \widetilde{q}_2; C[P]\kappa_2), (\sigma; \widetilde{q}_2; C[Q]\kappa_2)) \cup \mathcal{R}$. We have the transitions $(\sigma; \widetilde{q}_2; (C[P])\kappa_2) \xrightarrow{c?[\widetilde{v}_1, \widetilde{q}_1]} (\sigma; \widetilde{q}_1, \widetilde{q}_2; C''[P\kappa_2\kappa_1])$ and $(\sigma; \widetilde{q}_2; (C[Q])\kappa_2) \xrightarrow{c?[\widetilde{v}_1], \widetilde{q}_1} (\sigma; \widetilde{q}_1, \widetilde{q}_2; C''[Q\kappa_2, \kappa_1])$ where $C'' = C'\kappa_2\kappa_1$. Because $\widetilde{x}_1$ are bound in $C[P]$ and $C[Q]$, we have that $\widetilde{x}_1$ and $\widetilde{x}_2$ are distinct. The inductive hypothesis gives $((\sigma; \widetilde{q}_1, \widetilde{q}_2; C''[P\kappa_2\kappa_1]), (\sigma; \widetilde{q}_1, \widetilde{q}_2; C''[Q\kappa_2, \kappa_1])) \in \mathcal{R}$, therefore $\mathcal{R}'$ is a probabilistic branching bisimulation.

Output: Let $C = c![\widetilde{x}_1, \widetilde{x}_2].C'$ and $\kappa_1 = \{\widetilde{v_1}, \widetilde{q_1}/\widetilde{x_1}\}$ and $\kappa_2 = \{\widetilde{v}_2, \widetilde{q}_2/\widetilde{x}_2\}$ and $\kappa = \kappa_1\kappa_2$. Then let $\mathcal{R}' = ((\sigma; \widetilde{q}_1, \widetilde{q}_2; C[P]\kappa), (\sigma; \widetilde{q}_1, \widetilde{q}_2; C[Q]\kappa)) \cup \mathcal{R}$. If $\widetilde{x_2}$ is not empty, then $(\sigma; \widetilde{q}_1, \widetilde{q}_2; (C[P])\kappa)$ and $(\sigma; \widetilde{q}_1, \widetilde{q}_2; (C[Q])\kappa)$ have no transitions. If $\widetilde{x}_2$ is empty, then we have the transitions $(\sigma; \widetilde{q_1}, \widetilde{q}_2; (C[P])\kappa) \xrightarrow{c![\widetilde{v}_1]} (\sigma; \widetilde{q}_2; C''[P\kappa])$ and $(\sigma; \widetilde{q}_1, \widetilde{q}_3; (C[Q])\kappa) \xrightarrow{c![\widetilde{v}_1]} (\sigma; \widetilde{q}_2; C''[Q\kappa])$ where $C'' = C'\kappa$. Using IT-Out and Lemma 3.10 gives $\widetilde{y} \notin fv(C'[P]), fv(C'[Q])$, therefore $(C'[P])\kappa = (C'[P])\kappa_1'\kappa_2$ and $(C'[Q'])\kappa = (C'[Q])\kappa_1'\kappa_2$ where $\kappa_1' = \{\widetilde{v_1}/\widetilde{q_1}\}$. The inductive hypothesis gives $(\sigma; \widetilde{q_2}; (C'[P])\kappa_1'\kappa_2) \leftrightarrows (\sigma; \widetilde{q_2}; (C'[Q])\kappa_1'\kappa_2)$. Therefore $(\sigma; \widetilde{q_1}, \widetilde{q_2}; C[P]\kappa) \leftrightarrows (\sigma; \widetilde{q_1}, \widetilde{q_2}; C[Q]\kappa)$.

Restriction: Let $C = (\nu\widetilde{c})C'[]$ and $\kappa = \{\widetilde{v}/\widetilde{x}\}$ and $\widetilde{q}$ are the qubit names in $\widetilde{v}$. Define a relation

$$\mathcal{R}' = \{((\sigma_!; \omega_1; C[P]\kappa), (\sigma_2; \omega_2; C[Q]\kappa)) \mid (\sigma_1; \omega_1; C'[P]\kappa) \leftrightarrows (\sigma_2; \omega_2; C'[Q]\kappa)\} \ .$$

Then we have the derivation

$$\frac{(\sigma_1; \omega_1; (C'[P])\kappa) \xrightarrow{\alpha} (\sigma_1'; \omega_1'; C'[P'])}{(\sigma_1; \omega_1; (C[P])\kappa) \xrightarrow{\alpha} (\sigma_1'; \omega_1'; C[P'])}$$

Then we have

$$(\sigma_2; \omega_2; C'[Q]) \Longrightarrow (\sigma_2'; \omega_2'; C'[Q']) \xrightarrow{\alpha} (\sigma_2''; \omega_2''; C'[Q''])$$

where $(\sigma_1; \omega_1; C'[P]) \leftrightarrows (\sigma_2'; \omega_2'; C'[Q'])$ and $(\sigma_1'; \omega_1'; C'[P']) \leftrightarrows (\sigma_2''; \omega_2''; C'[Q''])$. By L-Res we have $(\sigma_2; \omega_2; C[Q]) \Longrightarrow (\sigma_2'; \omega_2'; C[Q']) \xrightarrow{\alpha} (\sigma_2''; \omega_2''; C[Q''])$, and we have $((\sigma_1; \omega_1; C[P]), (\sigma_2'; \omega_2'; C[Q'])) \in \mathcal{R}'$ and $((\sigma_1'; \omega_1'; C[P']), (\sigma_2''; \omega_2''; C[Q''])) \in \mathcal{R}'$.

Qubit declaration: Let $C = (\mathsf{qbit}\ x)C'$ and $\kappa_1 = \{\widetilde{v_1}/\widetilde{x_1}\}$ then $(\sigma; \widetilde{q}; (C[P])\kappa_1) \xrightarrow{\tau} (\sigma'; \widetilde{q}, q; C''[P\kappa_1\kappa_2])$ and $(\sigma; \widetilde{q}; (C[Q])\kappa_1) \xrightarrow{\tau} (\sigma'; \widetilde{q}, q; C''[Q\kappa_1\kappa_2])$ where $\kappa_2 = \{q/x\}$ and $C'' = C'\kappa_1\kappa_2$. Applying the inductive hypothesis gives $(\sigma'; \widetilde{q}, q; C''[P\kappa_1\kappa_2]) \leftrightarrows (\sigma'; \widetilde{q}, q; C''[Q\kappa_1\kappa_2])$ therefore $(\sigma; \widetilde{q}; (C[P])\kappa_1) \leftrightarrows (\sigma; \widetilde{q}; (C[Q])\kappa_1)$. □

3.4.1 Parallel Preservation

The existing bisimilarities that have been defined for QPAlg and qCCS are not congruence relations for general quantum processes. In particular, these relations are not preserved by parallel composition. Examples 3.2 and 3.3 demonstrate that our probabilistic branching bisimilarity is also not preserved by parallel composition. While the processes clearly result in different quantum states, their external behaviour is identical, and therefore they are considered bisimilar.

Example 3.2. Consider processes P and Q, where

$$P = c?[x].\{\mathsf{measure}\ x\}.\mathbf{0}$$
$$Q = c?[x].\{x *= \mathsf{H}\}.\{\mathsf{measure}\ x\}.\mathbf{0}$$

Define an equivalence relation $\mathcal{R}$ such that, for all σ, σ', $((\sigma; \emptyset; T), (\sigma; \emptyset; U)) \in \mathcal{R}$, $((\sigma; q; \{\mathsf{measure}\ q\}.\mathbf{0}), (\sigma'; q; \{\mathsf{measure}\ q\}.\mathbf{0})) \in \mathcal{R}$ and $((\sigma; q; \{\mathsf{measure}\ q\}.\mathbf{0}), (\sigma'; \{q *= \mathsf{H}\}.\{\mathsf{measure}\ q\}.\mathbf{0})) \in \mathcal{R}$. Then $\mathcal{R}$ is a probabilistic branching bisimulation, hence $P \leftrightarrows Q$.

On condition that we remember it is only the observable behaviour that is of interest, there should be no problem accepting the equivalence of P and Q. The issue with this arises when the processes are considered in a context; specifically the effect of entanglement means that these processes are not congruent. This is illustrated by the following example in which qubits p and q are entangled, therefore the measurement of p affects the state of q, resulting in different possibilities in each case for ρ^q when the action $d![q]$ occurs.

Example 3.3. Let $C = d![y].\mathbf{0} \parallel []$. Consider the entangled state $[pq \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]$, then the configurations corresponding to $C[P]$ and $C[Q]$ are

$$s = ([|pq\rangle \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; c, d; d![q].\mathbf{0} \parallel c?[x].\{x *= \mathsf{H}\}.\mathsf{measure}\ x.\mathbf{0})$$
$$t = ([|pq\rangle \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; c, d; d![q].\mathbf{0} \parallel c?[x].\mathsf{measure}\ x.\mathbf{0})$$

In the execution of s, the possible values of ρ^q when the action $d![q]$ occurs are $\frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|)$, $|0\rangle\langle 0|$, and $|1\rangle\langle 1|$, while in the execution of t, the possibilities are $\frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|)$, $\frac{|0\rangle\langle 0|+|0\rangle\langle 1|+|1\rangle\langle 0|+|1\rangle\langle 1|}{2}$ and $\frac{|0\rangle\langle 0|-|0\rangle\langle 1|-|1\rangle\langle 0|+|1\rangle\langle 1|}{2}$. The difference in the last two cases is a result of the Hadamard operation that was applied prior to measurement.

Despite the fact that bisimilarity is not preserved by parallel composition, it is still interesting to consider the effects of parallel contexts on the teleportation protocol. To this end, we now work towards proving Theorem 3.39, which states that equivalence

to *QChannel* is preserved by *all contexts*. Although this is not the congruence result that we would hope for, it does provide significant insight into the role of measurement and entanglement in parallel processes. In Section 3.5, we will discuss the implications further.

We prove this theorem through a series of Lemmas. Lemmas 3.25, 3.26, and 3.27 identify the structure of the bisimulation relating *QChannel* and a bisimilar process Q. In particular, for a given quantum state we can define a bisimulation with three equivalence classes. This is a generalisation of the bisimulations that were used in Section 3.3. These results are used later to determine the possible actions of the respective processes in the parallel context. The second stage involves determining the quantum state when the observable actions occur. Lemmas 3.31 and 3.33 prove that the respective reduced density operators of the input and output qubits are identical.

Lemma 3.24. *If t is a configuration with no τ-transitions and $t \leftrightarrows s$ and $s \Longrightarrow s'$ where $s' = \boxplus_i\ p_i \bullet s_i$ then $t \leftrightarrows s'$.*

Proof. By induction on the length of the sequence of transitions $s \Longrightarrow s'$. If $s_k \xrightarrow{\tau} s_{k+1}$ then applying the inductive hypothesis gives $t \leftrightarrows s_k$, therefore there exist configurations t', t'' such that $t \Longrightarrow t' \xrightarrow{\tau}{}^{+} t''$ where $s_k \leftrightarrows t'$ and $s_{k+1} \leftrightarrows t''$. The only configuration satisfying these conditions is t since t admits no τ-transitions, therefore $s_{k+1} \leftrightarrows t$. Else if $s_k \overset{p}{\rightsquigarrow} s_{k+1}$ then $s_{k+1} \leftrightarrows t$ because $t \leftrightarrows s_k$ implies $\mu(t, [t]_{\leftrightarrows}) = \mu(s_k, [t]_{\leftrightarrows})$ and by definition $\mu(t, [t]_{\leftrightarrows}) = 1$. □

Lemma 3.25. *If $QChannel \leftrightarrows Q$ and $(\sigma; \emptyset; Q) \Longrightarrow \boxplus_i\ p_i \bullet (\sigma_i; \omega_i; Q_i)$ then $(\sigma; \emptyset; QChannel) \leftrightarrows \boxplus_i\ p_i \bullet (\sigma_i; \omega_i; Q_i)$.*

Proof. The configuration $(\sigma; \emptyset; QChannel)$ admits no τ-transitions, therefore the result follows from Lemma 3.24. □

Lemma 3.26. *If $QChannel \leftrightarrows Q$ and $(\sigma; \emptyset; Q) \overset{c?[p]}{\Longrightarrow} \boxplus_i\ p_i \bullet (\sigma_i; \omega_i; Q_i)$ then $(\sigma; p; d![p].\mathbf{0}) \leftrightarrows \boxplus_i\ p_i \bullet (\sigma_i; \omega_i; Q_i)$.*

Proof. By definition there is a sequence of transitions

$$(\sigma; \emptyset; Q) \Longrightarrow (\sigma_1; \omega_1; Q_1) \overset{c?[p]}{\longrightarrow} (\sigma_2; \omega_2; Q_2) \Longrightarrow \boxplus_i\ p_i \bullet (\sigma_i; \omega_i; Q_i)$$

By Lemma 3.25 we have $(\sigma; \emptyset; QChannel) \leftrightarrows (\sigma_1; \omega_1; Q_1)$. It follows from the definition of bisimiliarity that $(\sigma; p; d![p].\mathbf{0}) \leftrightarrows (\sigma_2; \omega_2; Q_2)$. By Lemma 3.24 we obtain $(\sigma; p; d![p].\mathbf{0}) \leftrightarrows \boxplus_i\ p_i \bullet (\sigma_i; \omega_i; Q_i)$. □

Lemma 3.27. *If $QChannel \leftrightarrows Q$ and $(\sigma; \emptyset; Q) \overset{c?[p]}{\Longrightarrow}\overset{d![x]}{\Longrightarrow} \boxplus_i\ p_i \bullet (\sigma_i; \sigma_i; Q_i)$ then $(\sigma; \emptyset; \mathbf{0}) \leftrightarrows \boxplus_i\ p_i \bullet (\sigma_i; \sigma_i; Q_i)$.*

Proof. By definition there is a sequence of configurations

$$(\sigma;\emptyset;Q) \stackrel{c?[p]}{\Longrightarrow} (\sigma_1;\omega_1;Q_1) \stackrel{d![q]}{\longrightarrow} (\sigma_2;\omega_2;Q_2) \Longrightarrow \boxplus_i\ p_i \bullet (\sigma_i;\omega_i;Q_i)$$

By Lemma 3.26 we have $(\sigma_1;\omega_1;Q_1) \leftrightarrows (\sigma;p;d![p].\mathbf{0})$. It follows from the definition of bisimilarity that $(\sigma_2;\omega_2;Q_2) \leftrightarrows (\sigma;\emptyset;\mathbf{0})$. By Lemma 3.24 we obtain $(\sigma;\emptyset;\mathbf{0}) \leftrightarrows \boxplus_i\ p_i \bullet (\sigma_i;\omega_i;Q_i)$. □

We now define the equivalence relation $\mathcal{R}$, which we shall prove is a bisimulation. For all processes Q that are bisimilar to *QChannel*, this relation identifies all pairs of configurations that arise from $Q \parallel R$ through the same external actions. The relation is defined in terms of parameterised equivalence classes. These parameters include the quantum state and the context. We must consider cases in which the context creates a probabilistic distribution, hence the parameter may be a set $\{(p_i,\sigma_i,R_i)\}_i$, in which $\sum_i p_i = 1$. We also extend this relation to arbitrary contexts through the equivalence classes $D_0(\{(p_i,\sigma_i,C_i)\})$. In these classes, the contexts C_i are not parallel of the form $(\nu\widetilde{c})([]\parallel R)$, and hence contain configurations in which Q is guarded by a prefix.

Definition 3.9. For all Q and R such that $Q \leftrightarrows^c$ *QChannel* and $\Gamma_1;\emptyset \vdash Q$ and $\Gamma_2;\omega_r \vdash R$ and $\Gamma_1+\Gamma_2;\omega_r \vdash Q \parallel R$. Then

- $s \in D_0(\{(p_i,\sigma_i,C_i)\})$ if $s = \boxplus_i\ p_i \bullet (\sigma_i;\omega;C_i[Q])$ and $C_i \notin \cup_{\widetilde{c},R}\{(\nu\widetilde{c})([]\parallel R)\}$.
- $s \in D_1(\sigma,R,\widetilde{c})$ if $s = \boxplus_i\ p_i \bullet (\sigma_i;\omega,\omega_r;(\nu\widetilde{c})(R \parallel Q_i'))$ and $(\sigma;\emptyset;Q) \Longrightarrow \boxplus_i\ p_i \bullet (\sigma_i;\omega;Q_i')$.
- $s \in D_1(\{(p_i,\sigma_i,R_i)\},\widetilde{c})$ if $s = \boxplus_i\ p_i \bullet (\sigma_i';\omega,\omega_r;(\nu\widetilde{c})(R_i \parallel Q'))$ and $\forall i.(p_i < 1$ and $(\sigma_i;\emptyset;Q) \Longrightarrow (\sigma_i';\omega;Q'))$.
- $s \in D_2(\sigma,R,\widetilde{c})$ if $s = \boxplus_i\ p_i \bullet (\sigma_i;\omega,\omega_r;(\nu\widetilde{c})(R \parallel Q_i'))$ and $(\sigma;\emptyset;Q) \stackrel{c?[p]}{\Longrightarrow} \boxplus_i\ p_i \bullet (\sigma_i;\omega;Q_i)$.
- $s \in D_2(\{(p_i,\sigma_i,R_i)\},\widetilde{c})$ if $s = \boxplus_i\ p_i \bullet (\sigma_i';\omega,\omega_r;(\nu\widetilde{c})(R_i \parallel Q'))$ and $\forall i.(p_i < 1$ and $(\sigma_i;\emptyset;Q) \stackrel{c?[p]}{\Longrightarrow} (\sigma_i';\omega;Q'))$.
- $s \in D_3(\sigma,R,\widetilde{c})$ if $s = \boxplus_i\ p_i \bullet (\sigma_i;\omega,\omega_r;(\nu\widetilde{c})(R \parallel Q_i'))$ and $(\sigma;\emptyset;Q) \stackrel{c?[p]d![v]}{\Longrightarrow\Longrightarrow} \boxplus_i\ p_i \bullet (\sigma_i;\omega;Q_i)$.
- $s \in D_3(\{(p_i,\sigma_i,R_i)\},\widetilde{c})$ if $s = \boxplus_i\ p_i \bullet (\sigma_i';\omega,\omega_r;(\nu\widetilde{c})(R_i \parallel Q'))$ and $\forall i.(p_i < 1$ and $(\sigma_i;\emptyset;Q) \stackrel{c?[p]d![v]}{\Longrightarrow\Longrightarrow} (\sigma_i';\omega;Q'))$.

Then $(s,t) \in \mathcal{R}$ if there exists γ such that γ is a class defined above and $s,t \in \gamma$.

Lemmas 3.30 and 3.31 consider the ability of configurations in $D_1(\sigma, R, \widetilde{c})$ to match the input action $c?[p]$ of *QChannel*. This first lemma proves that the quantum state in $D_1(\sigma, R, \widetilde{c})$ is of the form $[\tilde{r}\tilde{q} \mapsto |\pi\rangle|\phi\rangle]$. This result is then used in Lemma 3.31 to determine the reduced density matrix ρ^p when the input occurs.

Lemma 3.28 (Preservation of seperable states)**.** *Let* $\sigma = [\widetilde{p}, \widetilde{q} \mapsto \sum_j |\phi_j\rangle|\psi_j\rangle]$. *If* $\Gamma_1; \widetilde{p} \vdash P$ *and* $(\sigma; \omega; P \parallel Q) \xrightarrow{\tau} \boxplus_i\ p_i \bullet (\sigma_i; \omega'; P_i \parallel Q)$ *then there exists* $\widetilde{p}', |\phi_{ij}\rangle$ *such that* $\Gamma; \widetilde{p}' \vdash P_i$ *and* $\sigma_i = [\widetilde{p}'\widetilde{q} \mapsto |\phi_{ij}\rangle|\psi_j\rangle]$.

Proof. By induction on the derivation of the transition. We consider the cases that alter the quantum state; the rest are straightforward.

L-QBIT: We have $([\widetilde{pq} \mapsto \sum_j |\phi_j\rangle|\psi_j\rangle]; \omega; P \parallel Q) \xrightarrow{\tau} ([\widetilde{p}p\widetilde{q} \mapsto |\phi'_j\rangle|\psi_j\rangle]; \omega, p; P' \parallel Q)$ where $|\phi'_j\rangle = |\phi_j\rangle|0\rangle$. By Theorem 3.14 we have $\Gamma_1; \widetilde{p}, p \vdash P'$.

R-TRANS: We have $([\widetilde{pq} \mapsto \sum_j |\phi_j\rangle|\psi_j\rangle]; \omega; \widetilde{r} *= U^m) \longrightarrow_v (\sigma'; \omega; \mathsf{unit})$. The typing derivation of $\Gamma_1; \widetilde{p} \vdash P$ gives $\widetilde{r} \subseteq \widetilde{p}$. Assume $\widetilde{p} = \widetilde{r}\widetilde{s}$ then $\sigma' = [\widetilde{r}\widetilde{s}\widetilde{q} \mapsto (U^m \otimes I_{\widetilde{s}} \otimes I_{\widetilde{q}})(\sum_j |\phi_j\rangle|\psi_j\rangle) = |\phi'_j\rangle|\psi_j\rangle]$ where $|\phi'_j\rangle = (U^m \otimes I_{\widetilde{s}})|\phi_j\rangle$. By Theorem 3.14 we have $\Gamma_1; \widetilde{p} \vdash P'$.

R-MEASURE: We have $([\widetilde{pq} \mapsto \sum_j |\phi_j\rangle|\psi_j\rangle]; \omega; \mathsf{measure}\ \widetilde{r}) \longrightarrow_v \boxplus_i\ p_i \bullet (\sigma_i; \omega; i)$. The typing derivation of $\Gamma_1; \widetilde{p} \vdash P$ gives $\widetilde{r} \subseteq \widetilde{p}$ therefore $\sigma_i = [\widetilde{pq} \mapsto \sum_j |\phi_{ij}\rangle|\psi_j\rangle]$ and by Theorem 3.14 we have $\Gamma_1; \widetilde{p} \vdash P_i$. □

Lemma 3.29 (Quantum state independence)**.** *If* $\Gamma_1; \widetilde{p} \vdash P$ *and* $P \parallel Q$ *is well-typed and* $([\widetilde{pq} \mapsto |\phi\rangle|\psi\rangle]; \omega_1, \omega_2; P \parallel Q) \xrightarrow{\alpha} \boxplus_i\ p_i \bullet ([\widetilde{p}'\widetilde{q} \mapsto |\phi_i\rangle|\psi\rangle]; \omega'_1, \omega_2; P_i \parallel Q)$ *and* $fq(P) \subseteq \omega_1$ *and* $fq(Q) \subseteq \omega_2$ *then for any* $Q', |\psi'\rangle$ *such that* $P \parallel Q'$ *is typed and* $fq(Q') \subseteq \omega'_2$, $([\widetilde{pq}' \mapsto |\phi\rangle|\psi'\rangle]; \omega_1, \omega'_2; P \parallel Q') \xrightarrow{\alpha} \boxplus_i\ p_i \bullet ([\widetilde{p}'\widetilde{q}' \mapsto |\phi_i\rangle|\psi'\rangle]; \omega'_1, \omega'_2; P_i \parallel Q')$.

Proof. The derivation of the transition using L-PAR gives $([\widetilde{pq} \mapsto |\phi\rangle|\psi\rangle]; \omega_1, \omega_2; P) \longrightarrow \boxplus_i\ p_i \bullet ([\widetilde{p}'\widetilde{q} \mapsto |\phi_i\rangle|\psi\rangle]; \omega'_1, \omega_2; P_i)$ hence for any Q' we obtain

$$([\widetilde{pq} \mapsto |\phi\rangle|\psi\rangle]; \omega_1, \omega_2; P \parallel Q') \longrightarrow \boxplus_i\ p_i \bullet ([\widetilde{p}'\widetilde{q} \mapsto |\phi_i\rangle|\psi\rangle]; \omega'_1, \omega_2; P_i \parallel Q')$$

The next condition $(\forall|\psi'\rangle)$ is proved by induction on the derivation of the transition. We consider the cases that alter the quantum state; the rest are straightforward.

L-QBIT: We have the transition $([\widetilde{pq} \mapsto |\phi\rangle|\psi\rangle]; \omega_1, \omega_2; P \parallel Q) \xrightarrow{\tau} ([\widetilde{p}p\widetilde{q} \mapsto |\phi'\rangle|\psi\rangle]; \omega'_1, \omega_2; P' \parallel Q)$ where $\omega'_1 = \omega_1, p$. Similarly, we also have the transition $([\widetilde{pq}' \mapsto |\phi\rangle|\psi'\rangle]; \omega_1, \omega'_2; P \parallel Q) \xrightarrow{\tau} ([\widetilde{p}p\widetilde{q}' \mapsto |\phi'\rangle|\psi'\rangle]; \omega'_1, \omega'_2; P' \parallel Q)$ where $\omega'_1 = \omega_1, p$.

R-TRANS: We have $([\widetilde{pq} \mapsto |\phi\rangle|\psi\rangle]; \omega_1, \omega_2; \widetilde{r} *= U^m) \longrightarrow_v (\sigma'; \omega_1, \omega_2; \mathsf{unit})$ where typing gives $\widetilde{r} \subseteq \widetilde{p}$ therefore $\sigma' = [\widetilde{r}\widetilde{s}\widetilde{q} \mapsto |\phi'\rangle|\psi\rangle]$ and $|\phi'\rangle = (U^m \otimes I_{\widetilde{s}})|\phi\rangle$. Because the identity transformation is applied to the rest of the qubits, we have $([\widetilde{pq}' \mapsto |\phi\rangle|\psi'\rangle]; \omega_1, \omega'_2; \widetilde{r} *= U^m) \longrightarrow_v (\sigma''; \omega_1, \omega'_2; \mathsf{unit})$ where $\sigma'' = [\widetilde{r}\widetilde{s}\widetilde{q}' \mapsto |\phi'\rangle|\psi'\rangle]$ and $|\phi'\rangle = (U^m \otimes I_{\widetilde{s}})|\phi\rangle$.

R-MEASURE: We have $([\widetilde{pq} \mapsto |\phi\rangle|\psi\rangle]; \omega_1, \omega_2; \mathsf{measure}\ \widetilde{r}) \longrightarrow_v \boxplus_i\ p_i \bullet (\sigma_i; \omega_1, \omega_2; i)$. The typing derivation gives $\widetilde{r} \subseteq \widetilde{p}$ therefore $\sigma_i = [\widetilde{pq} \mapsto |\phi_i\rangle|\psi\rangle]$. Similarly if $([\widetilde{pq'} \mapsto |\phi\rangle|\psi'\rangle]; \omega_1, \omega_2; \mathsf{measure}\ \widetilde{r}) \longrightarrow_v \boxplus_i\ p_i \bullet (\sigma_i; \omega_1, \omega_2; i)$ then $\sigma_i = [\widetilde{pq'} \mapsto |\phi_i\rangle|\psi'\rangle]$. □

Lemma 3.30 (Quantum state in D_1)**.** *If* $\Gamma; \emptyset \vdash Q$ *and* $\sigma = [\widetilde{r} \mapsto |\phi\rangle]$ *and* $(\sigma; \emptyset; Q) \Longrightarrow (\sigma'; \omega; Q')$ *then there exists* $\widetilde{q}$ *and* $|\psi\rangle$ *such that* $\sigma' = [\widetilde{rq} \mapsto |\phi\rangle|\psi\rangle]$ *and* $\forall |\phi'\rangle.([\widetilde{r'} \mapsto |\phi'\rangle]; \emptyset; Q) \Longrightarrow ([\widetilde{r'q} \mapsto |\phi'\rangle|\psi\rangle]; \omega; Q')$ *and* $\Gamma; \widetilde{q} \vdash Q'$.

Proof. By induction on the length of the sequence of transitions. We have $(\sigma_n; \omega_n; Q_n) \xrightarrow{\tau} (\sigma_{n+1}; \omega_{n+1}; Q_{n+1})$. The inductive hypothesis gives $\sigma_n = [\widetilde{rq}_n \mapsto |\phi\rangle|\psi_n\rangle]$ and $\Gamma; \widetilde{q}_n \vdash Q_n$ and $\forall |\phi'\rangle.(([\widetilde{r'} \mapsto |\phi'\rangle]; \emptyset; Q) \Longrightarrow ([\widetilde{r'q}_n \mapsto |\phi'\rangle|\psi_n\rangle]; \omega_n; Q_n))$. Lemma 3.28 gives $\sigma_{n+1} = [\widetilde{rq}_{n+1} \mapsto |\phi\rangle|\psi_{n+1}\rangle]$ and $\Gamma; \widetilde{q}_{n+1} \vdash Q_{n+1}$. Lemma 3.29 gives $\forall |\phi'\rangle.(([\widetilde{r'q}_n \mapsto |\phi'\rangle|\psi_n\rangle]; \omega_n; Q_n) \xrightarrow{\tau} ([\widetilde{r'q}_{n+1} \mapsto |\phi'\rangle|\psi_{n+1}\rangle]; \omega_{n+1}; Q_{n+1}))$. □

Lemma 3.31 (Input matching for $D_1(\sigma, R, \widetilde{c})$)**.** *If* $s \in \Gamma_1(\sigma, R, \widetilde{c})$ *and* $\sigma = [\widetilde{rp} \mapsto |\phi_0\rangle|0\rangle + |\phi_1\rangle|1\rangle]$ *and* $c \notin \widetilde{c}$ *and* $p \notin fq(R)$ *then there exist* s', s'' *such that* $s \Longrightarrow s' \xrightarrow{c?[p]} s''$ *and* $\rho^p = \sum_{i,j \in \{0,1\}} \langle \phi_i | \phi_j \rangle |i\rangle\langle j|$ *and* $s' \in D_1(\sigma, R, \widetilde{c})$ *and* $s'' \in D_2(\sigma, R, \widetilde{c})$.

Proof. Let $s = (\sigma_1; \omega_1, \omega_r; (\nu \widetilde{c})(Q_1 \parallel R)) \in D_1(\sigma, R, \widetilde{c})$. Then there exists a configuration $t = (\sigma_1; \omega_1; Q_1)$ such that $(\sigma; \emptyset; Q) \Longrightarrow t$. Lemma 3.25 gives $(\sigma; \emptyset; QChannel) \leftrightarrows t$, therefore there exist configurations $t' = (\sigma_1'; \omega_1'; Q_1')$ and $t'' = (\sigma_1''; \omega_1''; Q_1'')$ such that $t \Longrightarrow t' \xrightarrow{c?[p]} t''$ and $\rho^p = \sum_{i,j \in \{0,1\}} \langle \phi_i | \phi_j \rangle |i\rangle\langle j|$. Using L-PAR and L-RES and Lemma 3.29 on each transition in the sequence we obtain the required sequence of transitions $s \Longrightarrow s' \xrightarrow{c?[p]} s''$ where $s' = (\sigma_1'; \omega_1', \omega_r; (\nu \widetilde{c})(Q_1' \parallel R))$ and $s'' = (\sigma_1''; \omega_1'', \omega_r; (\nu \widetilde{c})(Q_1'' \parallel R))$. By definition we have $s' \in D_1(\sigma, R, \widetilde{c})$ and $s'' \in D_2(\sigma, R, \widetilde{c})$. □

We now prove a similar result, this time for the state of the output qubit from $D_2(\sigma, R, \widetilde{c})$.

Lemma 3.32 (Quantum state in D_2)**.** *If* $\Gamma; \emptyset \vdash Q$ *and* $\sigma = [\widetilde{rp} \mapsto |\phi_0\rangle|0\rangle + |\phi_1\rangle|1\rangle]$ *and* $(\sigma; \emptyset; Q) \overset{c?[p]}{\Longrightarrow} (\sigma'; \omega'; Q')$ *then there exists* $|\psi_0\rangle, |\psi_1\rangle$ *such that* $\sigma' = [\widetilde{rq} \mapsto |\phi_0\rangle|\psi_0\rangle + |\phi_1\rangle|\psi_1\rangle]$ *and* $\Gamma; \widetilde{q} \vdash Q'$ *and* $\forall |\phi_0'\rangle, |\phi_1'\rangle.(([\widetilde{r'} \mapsto |\phi_0'\rangle|0\rangle + |\phi_1'\rangle|1\rangle]; \emptyset; Q) \overset{c?[p]}{\Longrightarrow} ([\widetilde{r'q} \mapsto |\phi_0'\rangle|\psi_0\rangle + |\phi_1'\rangle|\psi_1\rangle]; \omega'; Q'))$.

Proof. By induction on the length of the final sequence of τ-transitions. Base case: If $(\sigma; \emptyset; Q) \Longrightarrow (\sigma_0; \omega_0; Q_0) \xrightarrow{c?[p]} (\sigma_1; \omega, p; Q_1)$ then Lemma 3.30 gives $|\psi_0\rangle$ such that $\sigma_0 = [\widetilde{rp}\widetilde{q} \mapsto (|\phi_0\rangle|0\rangle + |\phi_1\rangle|0\rangle)|\psi_0\rangle]$ and $\Gamma; \widetilde{q} \vdash Q_0$. Let $|\psi_{00}\rangle = |0\rangle|\psi_0\rangle$ and $|\psi_{01}\rangle = |1\rangle|\psi_0\rangle$ then $\sigma_1 = \sigma_0 = [\widetilde{rp}\widetilde{q} \mapsto |\phi_0\rangle|\psi_{00}\rangle + |\phi_1\rangle|\psi_{01}\rangle]$. Theorem 3.14 gives $\Gamma; p, \widetilde{q} \vdash Q_1$.

Inductive step: We have $(\sigma_n; \omega_n; Q_n) \xrightarrow{\tau} (\sigma_{n+1}; \omega_{n+1}; Q_{n+1})$. The inductive hypothesis gives $\sigma_n = [\widetilde{rq}_n \mapsto |\phi_0\rangle|\psi_{n0}\rangle + |\phi_1\rangle|\psi_{n1}\rangle]$ and $\Gamma; \widetilde{q}_n \vdash Q_n$ and $\forall |\phi_0'\rangle, |\phi_1'\rangle.(([\widetilde{r'} \mapsto |\phi_0'\rangle|0\rangle + |\phi_1'\rangle|1\rangle]; \emptyset; Q) \Longrightarrow ([\widetilde{r'q}_n \mapsto |\phi_0'\rangle|\psi_{n0}\rangle + |\phi_1'\rangle|\psi_{n1}\rangle]; \omega_n; Q_n)$. Lemma 3.28 gives $\sigma_{n+1} = [\widetilde{rq}_{n+1} \mapsto |\phi_0\rangle|\psi_{n+1,0}\rangle + |\phi_1\rangle|\psi_{n+1,1}\rangle]$ and $\Gamma; \widetilde{q}_{n+1} \vdash Q_{n+1}$. Lemma 3.29 gives

$\forall|\phi_0'\rangle,|\phi_1'\rangle.(([\widetilde{r}'\widetilde{q}_n \mapsto |\phi_0'\rangle|\psi_{n0}\rangle + |\phi_1'\rangle|\psi_{n1}\rangle];\omega_n;Q_n) \xrightarrow{\tau} ([\widetilde{r}'\widetilde{q}_{n+1} \mapsto |\phi_0'\rangle|\psi_{n+1,0}\rangle + |\phi_1'\rangle|\psi_{n+1,1}\rangle];\omega_{n+1};Q_{n+1}))$. □

Lemma 3.33 (Output from $D_2(\sigma,R,\widetilde{c})$)**.** *If $s \in D_2(\sigma,R,\widetilde{c})$ and $\sigma = [\widetilde{r}p \mapsto |\phi_0\rangle|0\rangle + |\phi_1\rangle|1\rangle]$ and $c \notin \widetilde{c}$ then there exist s', s'' such that $s \Longrightarrow s' \xrightarrow{d![q]} s''$ and $s' \in D_2(\sigma,R,\widetilde{c})$ and $s'' \in D_3(\sigma,R,\widetilde{c})$ and $\rho^q = \sum_{i,j\in\{0,1\}}\langle\phi_i|\phi_j\rangle|i\rangle\langle j|$.*

Proof. Let $s = (\sigma_1;\omega_1,\omega_r;(\nu\widetilde{c})(Q_1 \parallel R))$. Then there exists a configuration $t = (\sigma_1;\omega_1;Q_1)$ such that $(\sigma;\emptyset;Q) \overset{c?[p]}{\Longrightarrow} t$ where $Q \Leftrightarrow QChannel$. Then Lemma 3.26 gives $(\sigma;\emptyset;d![p].\mathbf{0}) \Leftrightarrow t$, therefore there exist configurations $t' = (\sigma_1';\omega_1';Q_1')$ and $t'' = (\sigma_1'';\omega_1'';Q_1'')$ such that $t \Longrightarrow t' \xrightarrow{d![q]} t''$ and $\rho^q = \sum_{i,j\in\{0,1\}}\langle\phi_i|\phi_j\rangle|i\rangle\langle j|$. Using L-Par and L-Res we obtain the required sequence of transitions $s \Longrightarrow s' \xrightarrow{d![p]} s''$ where $s' = (\sigma_s';\omega_s',\omega_r;(\nu\widetilde{c})(Q_s' \parallel R)$ and $s'' = (\sigma_s'';\omega_s'',\omega_r;(\nu\widetilde{c})(Q_s'' \parallel R))$. By definition we have $s' \in D_2(\sigma,R,\widetilde{c})$ and $s'' \in D_3(\sigma,R,\widetilde{c})$. □

Lastly, Lemma 3.34 proves that the quantum state for configurations in $D_3(\sigma,R,\widetilde{c})$ is of the form $[\widetilde{r}\widetilde{q} \mapsto |\phi\rangle|\psi\rangle]$.

Lemma 3.34 (Quantum state in $D_3(\sigma,R,\widetilde{c})$)**.** *If $\Gamma;\widetilde{q} \vdash Q$ and $\sigma = [\widetilde{r} \mapsto |\phi\rangle]$ and $(\sigma;\emptyset;Q) \overset{c?[p]d![x]}{\Longrightarrow\Longrightarrow} (\sigma';\omega';Q')$ then there exists $|\psi\rangle$ such that $\sigma' = [\widetilde{r}\widetilde{q} \mapsto |\phi\rangle|\psi\rangle]$ and $\Gamma;\widetilde{q} \vdash Q'$ and $\forall|\phi'\rangle.(([\widetilde{r}' \mapsto |\phi'\rangle];\emptyset;Q) \overset{c?[p]d![x]}{\Longrightarrow\Longrightarrow} ([\widetilde{r}'\widetilde{q} \mapsto |\phi'\rangle|\psi\rangle];\omega';Q'))$.*

Proof. By induction on the length of the final sequence of τ-transitions. We have $(\sigma;\emptyset;Q) \overset{c?[p]}{\Longrightarrow} (\sigma';\omega';Q') \xrightarrow{d![q]} (\sigma_1;\omega_1;Q_1) \Longrightarrow (\sigma_n;\omega_n;Q_n)$.

Base case: Let $|\phi\rangle = |\phi_0\rangle|0\rangle + |\phi_1\rangle|1\rangle$ then (by Lemma 3.32) there exist $|\psi_0\rangle,|\psi_1\rangle$ such that $\sigma' = [\widetilde{r}\widetilde{q} \mapsto |\phi_0\rangle|\psi_0\rangle + |\phi_1\rangle|\psi_1\rangle]$ and $\Gamma;\widetilde{q} \vdash Q'$. Let $|\psi_0\rangle = |\psi_{00}\rangle|0\rangle + |\psi_{01}\rangle|1\rangle$ and $|\psi_1\rangle = |\psi_{10}\rangle|0\rangle + |\psi_{11}\rangle|1\rangle$ then $\sigma_1 = \sigma' = [\widetilde{r}x\widetilde{q}' \mapsto |\phi_0\rangle(|0\rangle|\psi_{00}\rangle + |1\rangle|\psi_{01}\rangle) + |\phi_1\rangle(|0\rangle|\psi_{10}\rangle + |1\rangle|\psi_{11}\rangle)]$. Then

$$\begin{aligned}
\rho^x =&(\langle\phi_0|\phi_0\rangle\langle\psi_{00}|\psi_{00}\rangle + \langle\phi_0|\phi_1\rangle\langle\psi_{00}|\psi_{10}\rangle + \langle\phi_1|\phi_0\rangle\langle\psi_{10}|\psi_{00}\rangle + \langle\phi_1|\phi_1\rangle\langle\psi_{10}|\psi_{10}\rangle)|0\rangle\langle 0|\\
&+(\langle\phi_0|\phi_0\rangle\langle\psi_{00}|\psi_{01}\rangle + \langle\phi_0|\phi_1\rangle\langle\psi_{00}|\psi_{11}\rangle + \langle\phi_1|\phi_0\rangle\langle\psi_{10}|\psi_{01}\rangle + \langle\phi_1|\phi_1\rangle\langle\psi_{10}|\psi_{11}\rangle)|0\rangle\langle 1|\\
&+(\langle\phi_0|\phi_0\rangle\langle\psi_{01}|\psi_{00}\rangle + \langle\phi_0|\phi_1\rangle\langle\psi_{01}|\psi_{10}\rangle + \langle\phi_1|\phi_0\rangle\langle\psi_{11}|\psi_{00}\rangle + \langle\phi_1|\phi_1\rangle\langle\psi_{11}|\psi_{10}\rangle)|1\rangle\langle 0|\\
&+(\langle\phi_0|\phi_0\rangle\langle\psi_{01}|\psi_{01}\rangle + \langle\phi_0|\phi_1\rangle\langle\psi_{01}|\psi_{11}\rangle + \langle\phi_1|\phi_0\rangle\langle\psi_{11}|\psi_{01}\rangle + \langle\phi_1|\phi_1\rangle\langle\psi_{11}|\psi_{11}\rangle)|1\rangle\langle 1|
\end{aligned}$$

Lemma 3.33 gives $\rho^x = \sum_{i,j\in\{0,1\}}\langle\phi_i|\phi_j\rangle|i\rangle\langle j|$ therefore by comparison of coefficients we obtain

$$\langle\psi_{00}|\psi_{00}\rangle = \langle\psi_{00}|\psi_{11}\rangle = \langle\psi_{11}|\psi_{00}\rangle = \langle\psi_{11}|\psi_{11}\rangle = 1$$

and

$$\langle\psi_{10}|\psi_{10}\rangle = \langle\psi_{01}|\psi_{01}\rangle = 0$$

Therefore

$$|\psi_{00}\rangle = |\psi_{11}\rangle \text{ and } |\psi_{01}\rangle = |\psi_{10}\rangle = 0$$

So $[\widetilde{r}x\widetilde{q}' \mapsto (|\phi_0\rangle|0\rangle + |\phi_1\rangle|1\rangle)|\psi_{00}\rangle = |\phi\rangle|\psi_{00}\rangle]$. A similar argument shows that $\forall|\phi'\rangle.(([\widetilde{r}' \mapsto |\phi'\rangle]; \emptyset; Q) \stackrel{c?[p]d![x]}{\Longrightarrow} ([\widetilde{r}'\widetilde{q} \mapsto |\phi'\rangle|\psi_{00}\rangle]; \omega_1; Q_1)$. It follows from Theorem 3.14 that $\Gamma; \widetilde{q}' \vdash Q_1$.

Inductive step: We have $(\sigma_n; \omega_n; Q_n) \xrightarrow{\tau} (\sigma_{n+1}; \omega_{n+1}; Q_{n+1})$. The inductive hypothesis gives $\sigma_n = [\widetilde{r}\widetilde{q}_n \mapsto |\phi\rangle|\psi_n\rangle]$ and $\Gamma; \widetilde{q}_n \vdash Q_n$ and $\forall|\phi'\rangle.(([\widetilde{r}' \mapsto |\phi'\rangle]; \emptyset; Q) \stackrel{c?[p]}{\Longrightarrow}\stackrel{d![p]}{\Longrightarrow} ([\widetilde{r}'\widetilde{q}_n \mapsto |\phi'\rangle|\psi_n\rangle]; \omega_n; Q_n)$. Lemma 3.28 gives $\sigma_{n+1} = [\widetilde{r}\widetilde{q}_{n+1} \mapsto |\phi\rangle|\psi_{n+1}\rangle]$ and $\Gamma; \widetilde{q}_{n+1} \vdash Q_{n+1}$. Lemma 3.29 gives for all $|\phi'\rangle$

$$(([\widetilde{r}'\widetilde{q}_n \mapsto |\phi'\rangle|\psi_n\rangle]; \omega_n; Q_n) \xrightarrow{\tau} ([\widetilde{r}'\widetilde{q}_{n+1} \mapsto |\phi'\rangle|\psi_{n+1}\rangle]; \omega_{n+1}; Q_{n+1})) \ .$$

□

We now work towards proving Theorem 3.39 by first proving that the equivalence relation defined by the classes $D_3(\sigma, R, \widetilde{c})$ is a bisimulation. Then we build the equivalence relation in stages by including configurations in the classes $D_2(\sigma, R, \widetilde{c})$, and so on until we have the complete equivalence relation as defined in Definition 3.9.

Lemma 3.35. *Let $\mathcal{R}$ be an equivalence relation defined by the equivalence classes $D_3(\sigma, R, \tilde{c})$. Then $\mathcal{R}$ is a bisimulation.*

Proof. By case analysis of the possible transitions. Let $s = (\sigma'; \omega', \omega_r; (\nu\widetilde{c})(R \parallel Q')) \in D_3(\sigma, R, \widetilde{c})$ and $\sigma = [\widetilde{r} \mapsto |\phi\rangle]$ then (Lemma 3.34) there exists $|\psi\rangle$ such that $\sigma' = [\widetilde{r}\widetilde{q} \mapsto |\phi\rangle|\psi\rangle]$ and $\omega' = \widetilde{q}, \omega_r$ and $\Gamma; \widetilde{q} \vdash Q'$.

1. Internal transition by Q': If $s \xrightarrow{\tau} s'$ where $s' = \boxplus_i \, p_i \bullet (\sigma_i; \omega'', \omega_r; (\nu\widetilde{c})(R \parallel Q_i))$ then the derivation (by L-Res and L-Par) has the hypothesis $(\sigma'; \omega', \omega_r; Q') \xrightarrow{\tau} \boxplus_i \, p_i \bullet (\sigma_i; \omega'', \omega_r; Q_i)$. Lemma 3.22 gives $(\sigma'; \omega'; Q') \xrightarrow{\tau} \boxplus_i \, p_i \bullet (\sigma_i; \omega''; Q_i)$ therefore $s' \in D_3(\sigma, R, \widetilde{c})$. For any $t \in D_3(\sigma, R, \widetilde{c})$ let $t = t' = t''$ then $t \Longrightarrow t' \xrightarrow{\tau}{}^{+} t''$ and $(s, t'), (s', t'') \in \mathcal{R}$.

2. Probabilistic transition by Q': If $s_i = (\sigma_i; \omega', \omega_r; (\nu\widetilde{c})(R \parallel Q_i))$ and $s = \boxplus_i \, p_i \bullet s_i$ then we have $(\sigma; \emptyset; Q) \stackrel{c?[p]}{\Longrightarrow}\stackrel{d![x]}{\Longrightarrow} \boxplus_i \, p_i \bullet (\sigma_i; \omega'; Q_i)$. We have $\boxplus_i \, p_i \bullet (\sigma_i; \omega'; Q_i) \overset{p_i}{\rightsquigarrow} (\sigma_i; \omega'; Q_i)$ therefore $\forall i.(s_i \in D_3(\sigma, R, \widetilde{c}))$ and $\mu(s, D_3(\sigma, R, \widetilde{c})) = 1$. Similarly for any $t \in \mathcal{S}_p \cap D_3(\sigma, R, \widetilde{c})$ we have $\mu(t, D_3(\sigma, R, \widetilde{c})) = 1$ and furthermore, for any $u \in \mathcal{S}_n \cap D_3(\sigma, R, \widetilde{c})$ we have by definition $\mu(u, D_3(\sigma, R, \widetilde{c})) = 1$.

3. Action by R: If $s' = (\sigma''; \omega', \omega_r'; (\nu\widetilde{c})(R' \parallel Q'))$ and $s \xrightarrow{\alpha} s'$ then (Lemma 3.28) $\sigma'' = [\widetilde{r}'\widetilde{q} \mapsto |\phi'\rangle|\psi\rangle]$. Using Lemma 3.29, for any $P', |\psi'\rangle$ we have $t = ([\widetilde{r}\widetilde{q}' \mapsto |\phi\rangle|\psi'\rangle]; \omega'', \omega_r; (\nu\widetilde{c})(R \parallel P'))$ and $t' = ([\widetilde{r}'\widetilde{q}' \mapsto |\phi'\rangle|\psi'\rangle]; \omega'', \omega_r'; (\nu\widetilde{c})(R' \parallel P'))$ and $t \xrightarrow{\alpha} t'$. $s, t \in D_3(\sigma, R, \widetilde{c})$ gives $([\widetilde{r} \mapsto |\phi\rangle]; \emptyset; Q) \stackrel{c?[p]}{\Longrightarrow}\stackrel{d![x]}{\Longrightarrow} ([\widetilde{r}\widetilde{q} \mapsto |\phi\rangle|\psi\rangle]; \omega'; Q')$ and $([\widetilde{r} \mapsto |\phi\rangle]; \emptyset; P) \stackrel{c?[p]}{\Longrightarrow}\stackrel{d![x]}{\Longrightarrow} ([\widetilde{r}\widetilde{q} \mapsto |\phi\rangle|\psi\rangle]; \omega'; P')$ and $\Gamma_R; \omega_r \vdash R$ and $\Gamma_1; \Sigma_1 \vdash R \parallel Q$ and $\Gamma_2; \Sigma_2 \vdash R \parallel P$. Lemma 3.29 gives $\Gamma_R; \omega_r' \vdash R'$ and Theorem 3.14 gives $\Gamma_1; \Sigma_1 \vdash R' \parallel Q$ and $\Gamma_2; \Sigma_2' \vdash R' \parallel P$. Therefore $s', t' \in D_3(\sigma, R', \widetilde{c})$.

4. Probabilistic transition by R: For any $s \in D_3(\{(p_i, \sigma_i, R_i)\}, \widetilde{c})$ then we have $\mu(s, D_3(\sigma_i, R_i, \tilde{c})) = p_i$.

□

Lemma 3.36. *The equivalence relation $\mathcal{R}$ consisting of the equivalence classes $D_3(\sigma, R, \tilde{c}) \cup D_2(\sigma, R, \tilde{c})$ is a bisimulation.*

Proof. By case analysis of the possible transitions.

1. Internal transition by Q': For $s = (\sigma'; \omega, \omega_r; (\nu\widetilde{c})(R \parallel Q'))$ we have $(\sigma; \emptyset; Q) \stackrel{c?[p]}{\Longrightarrow} (\sigma'; \omega; Q')$. If $s \stackrel{\tau}{\longrightarrow} s'$ where $s' = \boxplus_i\ p_i \bullet (\sigma_i; \omega', \omega_r; (\nu\widetilde{c})(R \parallel Q_i))$ then the derivation (L-Res and L-Par) gives $(\sigma'; \omega, \omega_r; Q') \stackrel{\tau}{\longrightarrow} \boxplus_i\ p_i \bullet (\sigma_i; \omega', \omega_r; Q_i)$. Theorem 3.14 gives $\Gamma; \omega \vdash Q'$ and Lemma 3.22 gives $(\sigma'; \omega; Q') \stackrel{\tau}{\longrightarrow} \boxplus_i\ p_i \bullet (\sigma_i; \omega'; Q_i)$. Therefore $s' \in D_2(\sigma, R, \widetilde{c})$. For any $t \in D_2(\sigma, R, \widetilde{c})$ let $t' = t'' = t$ then $t \Longrightarrow t' \stackrel{\tau}{\longrightarrow}^{+} t''$ and $(s, t), (s', t'') \in \mathcal{R}$.

2. Output by Q': If $\sigma = [\widetilde{r}p \mapsto |\phi_0\rangle|0\rangle = |\phi_1\rangle|1\rangle]$ and $s \stackrel{d![q]}{\longrightarrow} s'$ then Lemma 3.33 gives $s' \in D_3(\sigma, R, \widetilde{c})$ and $\rho^q = \sum_{i,j\in\{0,1\}} \langle\phi_i|\phi_j\rangle |i\rangle\langle j|$ and for any $t \in D_2(\sigma, R, \widetilde{c})$ there exists $t' \in D_2(\sigma, R, \widetilde{c})$ and $t'' \in D_3(\sigma, R, \widetilde{c})$ such that $t \Longrightarrow t' \stackrel{d![x]}{\longrightarrow}$ and $\rho^x = \sum_{i,j\in\{0,1\}} \langle\phi_i|\phi_j\rangle |i\rangle\langle j|$.

3. Communication between R and Q': Let $s = (\sigma_1; \omega_1, q, \omega_r; (\nu\widetilde{c})(R \parallel Q_1))$ and $s' = (\sigma_1; \omega_1, q, \omega_r; (\nu\widetilde{c})(R' \parallel Q_1'))$ then the derivation of the transition $s \stackrel{\tau}{\longrightarrow} s'$ has the hypotheses $(\sigma_1; \omega_1, \omega_r; R) \stackrel{d?[q]}{\longrightarrow} (\sigma_1; \omega_1, q, \omega_r; R')$ and $(\sigma_1; \omega_1, q, \omega_r; P) \stackrel{d![q]}{\longrightarrow} (\sigma_1; \omega_1, \omega_r; P')$. Lemma 3.33 gives for any $t \in D_2(\sigma, R, \widetilde{c})$ there exists t', t'' such that $t \Longrightarrow t' \stackrel{d![q]}{\longrightarrow} t''$ and $t' \in D_2(\sigma, R, \widetilde{c})$ and $t'' \in D_3(\sigma, R, \widetilde{c})$. The transition $t' \stackrel{d![q]}{\longrightarrow} t''$ has the hypothesis $(\sigma_2; \omega_2, q, \omega_r; Q_2) \stackrel{d![q]}{\longrightarrow} (\sigma_2; \omega_2, \omega_r; Q_2')$. Lemma 3.29 gives $(\sigma_2; \omega_2, \omega_r; R) \stackrel{d?[q]}{\longrightarrow} (\sigma_2; \omega_2, q, \omega_r; R')$ therefore (by L-Com and L-Res) $(\sigma_2; \omega_2, q, \omega_r; (\nu\widetilde{c})(R \parallel Q_2)) \stackrel{\tau}{\longrightarrow} (\sigma_2; \omega_2, q, \omega_r; (\nu\widetilde{c})(R' \parallel Q_2'))$. Furthermore we have (Theorem 3.14) $Q \parallel R'$ is typed and (by Lemma 3.22) $(\sigma_2; \omega_2, q; Q_2) \stackrel{d![q]}{\longrightarrow} (\sigma_2; \omega_2; Q_2')$. Therefore $(\sigma; \emptyset; Q) \stackrel{c?[p]\,d![q]}{\Longrightarrow\Longrightarrow} (\sigma_2; \omega_2; Q_2')$ gives $t'' \in D_3(\sigma, R', \widetilde{c})$.

4. Probabilistic transition by Q': If $s_i = (\sigma_i; \omega, \omega_r; (\nu\widetilde{c})(R \parallel Q_i))$ and $s = \boxplus_i\ p_i \bullet s_i$ then we have $(\sigma; \emptyset; Q) \stackrel{c?[p]}{\Longrightarrow} \boxplus_i\ p_i \bullet (\sigma_i; \omega; Q_i)$. Now $\boxplus_i\ p_i \bullet (\sigma_i; \omega; Q_i) \stackrel{p_i}{\rightsquigarrow} (\sigma_i; \omega; Q_i)$ therefore $\forall i.(s_i \in D_2(\sigma, R, \widetilde{c}))$ so $\mu(s, D_2(\sigma, R, \widetilde{c})) = 1$. Similarly for any $t \in \mathcal{S}_p \cap D_2(\sigma, R, \widetilde{c})$ we have $\mu(t, D_2(\sigma, R, \widetilde{c})) = 1$ and by definition, for any $u \in \mathcal{S}_n \cap D_2(\sigma, R, \widetilde{c})$ we have $\mu(u, D_2(\sigma, R, \widetilde{c})) = 1$.

5. Action by R: If $s = ([\tilde{r}\tilde{q} \mapsto |\phi_0\rangle|\psi_0\rangle + |\phi_1\rangle|\psi_1\rangle]; \omega, \omega_r; (\nu\widetilde{c})(R \parallel Q'))$ and $s' = ([\widetilde{r}'\widetilde{q} \mapsto |\phi_0'\rangle|\psi_0\rangle + |\phi_1'\rangle|\psi_1\rangle]; \omega, \omega_r'; (\nu\widetilde{c})(R' \parallel Q'))$ and $s \stackrel{\alpha}{\longrightarrow} s'$ then (Lemma 3.29) for any Q'' and $|\psi_i'\rangle$ we have $t = ([\widetilde{r}\widetilde{q}' \mapsto |\phi_0\rangle|\psi_0'\rangle + |\phi_1\rangle|\psi_1'\rangle]; \omega', \omega_r; (\nu\widetilde{c})(R \parallel Q''))$

and $t' = ([\widetilde{r}'\widetilde{q}' \mapsto |\phi_0'\rangle|\psi_0'\rangle + |\phi_1'\rangle|\psi_1'\rangle]; \omega', \omega_r'; (\nu\widetilde{c})(R' \parallel Q''))$ and $t \xrightarrow{\alpha} t'$. Because $s, t \in D_2(\sigma, R, \widetilde{c})$, we have $([\widetilde{r} \mapsto |\phi_0\rangle|0\rangle + |\phi_1\rangle|1\rangle]; \emptyset; Q) \overset{c?[p]}{\Longrightarrow} ([\widetilde{r}\widetilde{q} \mapsto |\phi_0\rangle|\psi_0\rangle + |\phi_1\rangle|\psi_1\rangle]; \omega; Q')$ and $([\widetilde{r} \mapsto |\phi_0\rangle|0\rangle + |\phi_1\rangle|1\rangle]; \emptyset; Q) \overset{c?[p]}{\Longrightarrow} ([\widetilde{r}\widetilde{q} \mapsto |\phi_0\rangle|\psi_0'\rangle + |\phi_1\rangle|\psi_1'\rangle]; \omega; Q'')$ and $\Gamma; \omega_r \vdash R \parallel Q$. Lemma 3.29 and Theorem 3.14 gives $\Gamma; \omega_r' \vdash R' \parallel Q$. Therefore $s', t' \in D_2([\widetilde{r}' \mapsto |\phi_0\rangle|0\rangle + |\phi_1\rangle|1\rangle], R', \widetilde{c})$.

6. Probabilistic transition by R: If $s \in D_2(\{p_i, \sigma_i, R_i\}, \widetilde{c})$ then $\mu(s, D_2(\sigma_i, R_i, \widetilde{c})) = p_i$.

□

Lemma 3.37. *The equivalence relation $\mathcal{R}$ consisting of the equivalence classes $D_3(\sigma, R, \tilde{c}) \cup D_2(\sigma, R, \tilde{c}) \cup D_1(\sigma, R, \tilde{c})$ is a bisimulation.*

Proof. By case analysis of the possible transitions. We prove that given each transition all other configurations are able to match it.

1. Internal transition by Q': For $s = (\sigma'; \omega, \omega_r; (\nu\widetilde{c})(R \parallel Q'))$ we have $(\sigma; \emptyset; Q) \Longrightarrow (\sigma'; \omega; Q')$. If $s \xrightarrow{\tau} s'$ where $s' = \boxplus_i\ p_i \bullet (\sigma_i; \omega', \omega_r; (\nu\widetilde{c})(R \parallel Q_i))$ then the derivation gives $(\sigma'; \omega, \omega_r; Q) \Longrightarrow \boxplus_i\ p_i \bullet (\sigma_i; \omega', \omega_r; Q_i)$. By Lemma 3.22 we have $(\sigma'; \omega; Q) \Longrightarrow \boxplus_i\ p_i \bullet (\sigma_i; \omega'; Q_i)$. Therefore $s' \in D_1(\sigma, R, \widetilde{c})$. For all $u \in D_1(\sigma, R, \widetilde{c})$ let $t' = t'' = t$ then $t \Longrightarrow t' \longrightarrow^+ t''$ and $(s, t'), (s', t'') \in \mathcal{R}$.

2. Input by Q': If $s \xrightarrow{c?[p]} s'$ then the matching transition follows from Lemma 3.31.

3. Communication between R and Q': Let $s = (\sigma_1; \omega_1, \omega_r, p; (\nu\tilde{c})R \parallel Q_1)$ and $s' = (\sigma_1; \omega_1, \omega_r, p; (\nu\tilde{c})R' \parallel Q_1')$, then the derivation (L-Res and L-Com) of the transition $s \xrightarrow{\tau} s'$ must contain the hypotheses $(\sigma_1; \omega_1, \omega_r, p; R) \xrightarrow{c![p]} (\sigma_1; \omega_1, \omega_r; R')$ and $(\sigma_1; \omega_1, \omega_r; Q_1) \xrightarrow{c?[p]} (\sigma_1; \omega, \omega_r, p; Q_1')$. We have $\Gamma; \omega_1 \vdash Q_1$ (Lemma 3.30) therefore by Lemma 3.22 we have $(\sigma_1; \omega_1; Q_1) \xrightarrow{c?[p]} (\sigma_1; \omega_1, p; Q_1')$. For any $t \in D_1(\sigma, R, \widetilde{c})$ we have $t = (\sigma_2; \omega_2, \omega_r, p; (\nu\widetilde{c})(R \parallel Q_2))$ and (by Lemma 3.25) $(\sigma_1; \omega_1; Q_1) \leftrightarrows (\sigma_2; \omega_2; Q_2)$. Therefore we have a sequence $(\sigma_2; \omega_2; Q_2) \Longrightarrow (\sigma_2'; \omega_2'; Q_2') \xrightarrow{c?[p]} (\sigma_2''; \omega_2''; Q_2'')$. Using Lemma 3.21 then L-Par and L-Res gives $t' = (\sigma_2'; \omega_2', \omega_r, p; (\nu\widetilde{c})(R \parallel Q_2'))$ and $t \Longrightarrow t'$. Then L-Com and L-Res give $t'' = (\sigma_2''; \omega_2'', \omega_r, p; (\nu\widetilde{c})(R' \parallel Q_2'))$ and $t' \xrightarrow{c?[p]} t''$. By definition we have $t' \in D_1(\sigma, R, \widetilde{c})$. There exists P and Q such that $(\sigma; \emptyset; Q) \Longrightarrow (\sigma_1; \omega_1; Q_1)$ and $(\sigma; \emptyset; P) \Longrightarrow (\sigma_2; \omega_2; Q_2)$ therefore Theorem 3.14 gives $\Gamma_1; \Sigma_1 \vdash R' \parallel Q$ and $\Gamma_2; \Sigma_2 \vdash R' \parallel P$, hence $s', t'' \in D_2(\sigma, R', \widetilde{c})$.

4. Probabilistic transition by Q': If $s_i = [\tilde{r}\tilde{q} \mapsto |\pi\rangle|\phi_i\rangle]; \omega, \omega_r; (\nu\tilde{c})R \parallel Q_i')$ and $s = \boxplus_i\ p_i \bullet s_i$ then we have $(\sigma; \emptyset; Q) \Longrightarrow \boxplus_i\ p_i \bullet (\sigma_i; \omega; Q_i)$. Now $\boxplus_i\ p_i \bullet (\sigma_i; \omega; Q_i) \overset{p_i}{\rightsquigarrow} (\sigma_i; \omega; Q_i)$ therefore $\forall i.(s_i \in D_1(\sigma, R, \widetilde{c}))$ so $\mu(s, D_1(\sigma, R, \tilde{c})) = 1$.

Similarly for any $t \in \mathcal{S}_p \cap D_1(\sigma, R, \tilde{c})$ we have $\mu(t, D_1(\sigma, R, \tilde{c})) = 1$ and by definition, for any $u \in \mathcal{S}_n \cap D_1(\sigma, R, \tilde{c})$ we have $\mu(u, D_1(\sigma, R, \tilde{c})) = 1$.

5. Action by R: If $s = ([\widetilde{r}\widetilde{q} \mapsto |\phi\rangle|\psi\rangle]; \omega, \omega_r; (\nu\widetilde{c})(R \parallel Q'))$ and $s \xrightarrow{\alpha} s'$ and $s' = ([\widetilde{r}'\widetilde{q} \mapsto |\phi'\rangle|\psi\rangle]; \omega, \omega_r'; (\nu\widetilde{c})(R' \parallel Q'))$ then (Lemma 3.29) for any Q'' and $|\psi'\rangle$ we have $t = ([\widetilde{r}\widetilde{q}' \mapsto |\phi\rangle|\psi'\rangle]; \omega', \omega_r; (\nu\widetilde{c})(R \parallel Q''))$ and $t \xrightarrow{\alpha} t'$ and $t' = ([\widetilde{r}'\widetilde{q}' \mapsto |\phi'\rangle|\psi'\rangle]; \omega', \omega_r'; (\nu\widetilde{c})(R' \parallel Q''))$. Because $s, t \in D_1([\widetilde{r} \mapsto |\phi\rangle], R, \widetilde{c})$, we have $([\widetilde{r} \mapsto |\phi\rangle]; \emptyset; Q) \Longrightarrow ([\widetilde{r}\widetilde{q} \mapsto |\phi\rangle|\psi\rangle]; \omega; Q')$ (respectively for Q''). Theorem 3.14 gives $\Gamma; \Sigma \vdash R' \parallel Q$. Therefore $s', t' \in D_1([\widetilde{r}' \mapsto |\phi'\rangle], R', \widetilde{c})$.

6. Probabilistic transition by R: For any $s \in D_1(\{p_i, \sigma_i, R_i\}, \tilde{c})$, by definition, we have $\mu(s, D_1(\sigma_i, R_i, \tilde{c})) = p_i$.

□

Lemma 3.38. *Let $s = (\sigma; \omega; C[P])$ and $t = (\sigma; \omega; C[Q])$ where $s, t \in D_0(1, \sigma, C)$. If $s \xrightarrow{\alpha} s'$ then either there exists C_i, κ, σ_i, ω' such that $s' = \boxplus_i\, p_i \bullet (\sigma_i; \omega'; C_i[P\kappa]$ and $t' = \boxplus_i\, p_i \bullet (\sigma_i; \omega'; C_i[Q\kappa])$ and $t \xrightarrow{\alpha} t'$, or $t \xrightarrow{\alpha} t'$ and $s' = t'$.*

Proof. By induction on the derivation of $s \xrightarrow{\alpha} s'$.

L-Out: We have $C = c![\widetilde{v}, \widetilde{q}].C'$, then $(\sigma; \omega, \widetilde{q}; C[P]) \xrightarrow{c![\widetilde{v},\widetilde{q}]} (\sigma; \omega; C'[P])$ and $(\sigma; \omega, \widetilde{q}; C[Q]) \xrightarrow{c![\widetilde{v},\widetilde{q}]} (\sigma; \omega; C'[Q])$.

L-In: We have $C = c?[\widetilde{x}].C'$. Then we have $(\sigma; \omega; C[P]) \xrightarrow{c?[\widetilde{v},\widetilde{q}]} (\sigma; \omega, \widetilde{q}; C''[P\kappa])$ and $(\sigma; \omega; C[Q]) \xrightarrow{c?[\widetilde{v},\widetilde{q}]} (\sigma; \omega, \widetilde{q}; C''[Q\kappa])$ where $C'' = C'\kappa$.

L-Com: There are three cases. Suppose $C = c![\widetilde{v}].R_1 \parallel c?[\widetilde{x}].R_2 \parallel C'$ then we have the transitions $(\sigma; \omega; C[P]) \xrightarrow{\tau} (\sigma; \omega; C''[P])$ and $(\sigma; \omega; C[Q]) \xrightarrow{\tau} (\sigma; \omega; C''[Q])$ where $C'' = R_1 \parallel R_2\{\widetilde{v}/\widetilde{x}\} \parallel C'$. Suppose $C = c![\widetilde{v}].C' \parallel c?[\widetilde{x}].R_1 \parallel R_2$ then $(\sigma; \omega; C[P]) \xrightarrow{\tau}$ $(\sigma; \omega; C''[P])$ and $(\sigma; \omega; C[Q]) \xrightarrow{\tau} (\sigma; \omega; C''[Q])$ where $C'' = C' \parallel R_1 \parallel R_2$. Suppose $C = c![\widetilde{v}, \widetilde{q}].R_1 \parallel c?[\widetilde{x}].C' \parallel R_2$ then $(\sigma; \omega; C[P]) \xrightarrow{\tau} (\sigma; \omega; C''[P\kappa])$ and $(\sigma; \omega; C[Q]) \xrightarrow{\tau}$ $(\sigma; \omega; C''[Q\kappa])$ where $\kappa = \{\widetilde{v}/\widetilde{x}\}$ and $C'' = R_1 \parallel C'\kappa \parallel R_2$.

L-Sum: We have $C = C' + R$ where C' is not empty. If $(\sigma; \omega; R) \xrightarrow{\alpha} (\sigma'\omega'; R')$ then (by L-Sum) we have $(\sigma; \omega; C[P]) \xrightarrow{\alpha} (\sigma'; \omega'; R')$ and $(\sigma; \omega; C[Q]) \xrightarrow{\alpha} (\sigma'; \omega'; R')$. The induction hypothesis gives $(\sigma; \omega; C'[P]) \xrightarrow{\alpha'} (\sigma''; \omega''; C''[P\kappa])$ and $(\sigma; \omega; C'[Q]) \xrightarrow{\alpha'}$ $(\sigma''; \omega''; C''[Q\kappa])$ therefore $(\sigma; \omega; C[P]) \xrightarrow{\alpha'} (\sigma''; \omega''; C''[P\kappa])$ and $(\sigma; \omega; C[Q]) \xrightarrow{\alpha'}$ $(\sigma''; \omega''; C''[Q\kappa])$,

L-Par: We have $C = C' \parallel R$ where C' is not empty $C' \neq (\nu c)[]$. Then $(\sigma; \omega; C[P]) \xrightarrow{\alpha} (\sigma'; \omega'; C''[P\kappa])$ has the hypothesis $(\sigma; \omega; C'[P]) \xrightarrow{\alpha} (\sigma'; \omega'; C'''[P\kappa])$ where $C'' = C''' \parallel R$. The inductive hypothesis gives $(\sigma; \omega; C'[Q]) \xrightarrow{\alpha} (\sigma'; \omega'; C'''[Q\kappa])$ therefore $(\sigma; \omega; C[Q]) \xrightarrow{\alpha} (\sigma'; \omega'; C''[Q\kappa])$. We also have $(\sigma; \omega; C[P]) \xrightarrow{\alpha'} (\sigma''; \omega''; C''''[P])$ and $(\sigma; \omega; C[Q]) \xrightarrow{\alpha'} (\sigma''; \omega''; C''''[Q])$ where $C'''' = C' \parallel R'$.

L-Res: We have $C = (\nu\widetilde{c})C'$ where C' is not empty and $C' \neq [] \parallel R$. If $(\sigma; \omega; C[P]) \xrightarrow{\alpha} (\sigma'; \omega'; C''[P\kappa])$ then we have $(\sigma; \omega; C'[P]) \xrightarrow{\alpha} (\sigma'; \omega'; C'''[P\kappa])$

where $C'' = (\nu\widetilde{c})C'''$. Applying the inductive hypothesis and L-RES gives $(\sigma;\omega;C[Q]) \xrightarrow{\alpha} (\sigma';\omega';C''[Q\kappa])$.

L-QBIT: We have $C = (\mathsf{qbit}\ x)C'$. If $(\sigma;\omega;C[P]) \xrightarrow{\tau} (\sigma';\omega,q;C'\kappa[P\kappa])$ where $\kappa = \{q/x\}$ then $(\sigma;\omega;C[Q]) \xrightarrow{\tau} (\sigma';\omega,q;C'\kappa[Q\kappa])$.

L-EXPR: The transition $(\sigma;\omega;C[P]) \xrightarrow{\tau} \boxplus_i\ p_i \bullet (\sigma_i;\omega;C_i[P])$ has the hypothesis $(\sigma;\omega;e) \xrightarrow{\tau} \boxplus_i\ p_i \bullet (\sigma_i;\omega;e_i)$ where $C[P] = F[e]$. By case analysis of the structure of F we find that P must appear complete in F, therefore $C_i[P] = F[e_i]$ and we can replace P by Q to get F' such that $C[Q] = F'[e]$ and $C_i[Q] = F'[e_i]$. Therefore $(\sigma;\omega;C[Q]) \xrightarrow{\tau} \boxplus_i\ p_i \bullet (\sigma_i;\omega;C_i[Q])$. □

Theorem 3.39. *If $\Gamma \vdash Q$ and $QChannel \leftrightarrows Q$, then for any context C such that $\Gamma \vdash C[QChannel]$ and $\Gamma \vdash C[Q]$, $C[QChannel] \leftrightarrows C[Q]$.*

Proof. Let $\mathcal{R}$ be an equivalence relation defined by the equivalence classes $D_0(\sigma, C)$, $D_1(\sigma, R, \tilde{c})$, $D_2(\sigma, R, \tilde{c})$, and $D_3(\sigma, R, \tilde{c})$.

If $s, t \in D_1(1, \sigma, C)$ then (by Lemma 3.38) if $s = (\sigma;\omega;C[P])$ and $t = (\sigma;\omega;C[Q])$ then $s \xrightarrow{\alpha} s'$ then $t \xrightarrow{\alpha} t'$ and either $s' = t'$ or $s' = \boxplus_i\ p_i \bullet (\sigma_i;\omega';C_i[P])$ and $t' = \boxplus_i\ p_i \bullet (\sigma_i;\omega';C_i[Q])$. Because $fv(QChannel) = \emptyset$ we have $\forall\kappa.(Q\kappa \leftrightarrows (QChannel)\kappa = QChannel)$. If $s' = t'$ then $(s', t') \in \mathcal{R}$. For a non-probabilistic state, if $C_i = (\nu\widetilde{c})([]\ \|\ R_i)$ then $s', t' \in D_1(\sigma_i, R, \widetilde{c})$. For a probabilistic state, if $\forall i.(C_i = (\nu\widetilde{c})([]\ \|\ R_i))$ then $s', t' \in D_1(\{p_i, \sigma_i, R_i\}, \widetilde{c})$. Otherwise $s', t' \in D_0(\{(p_i, \sigma_i, C_i)\})$. Therefore $(s', t') \in \mathcal{R}$.

If $s, t \in D_0(\{(p_i, \sigma_i, C_i)\})$ then $\forall i.(\mu(s, D_0(1, \sigma_i, C_i)) = p_i = \mu(t, D_0(1, \sigma_i, C_i)))$. □

3.5 Discussion

In this section, we present an analysis of the labelled transition system and the probabilistic bisimilarity defined in this chapter. We worked towards proving that quantum teleportation is bisimilar to a direct quantum channel. This equivalence is well-motivated and is one that has been considered in other formalisms, for example by Danos et al. [2007a]; D'Hondt [2005] and Abramsky and Coecke [2004]. The result that the equivalence of teleportation and a quantum channel is preserved by all contexts is in agreement with a similar result using measurement calculus [Danos et al. 2007a]. Because CQP is designed to model arbitrary implementations of communication protocols, the result is arguably stronger in this setting than it is for the measurement calculus.

Preservation by all contexts is a highly desirable property for a process equivalence, since it is the gateway to compositional analysis. Congruence relations have been sought by Lalire [2006], Feng et al. [2006] and Ying et al. [2007], however they have

not been successful for general quantum processes. Lalire [2006] gives two reasons that parallel preservation fails; the first is due to the control of names, which the type system in CQP controls in a distributed manner, and the second is a result of treating non-determinism as equi-probability. This second point is mitigated by our choice of probabilistic function μ in the definition of bisimilarity, which separates non-determinism from probabilistic transitions.

The equivalence defined in this chapter is not preserved by all contexts, as demonstrated by Examples 3.2 and 3.3. The component-oriented approach to modelling systems is illustrated by the processes in Example 3.2; the input action defines the external interface of these processes, and the absence of any output action indicates that these processes provide no output. The equivalence of these processes demonstrates both the observational and abstract aspects of our bisimilarity, since only the external state is considered relevant, while the internal behaviours are clearly different. Example 3.3 gives a process context that, according to this bisimilarity, is able to distinguish the processes of Example 3.2. Intuitively, it should not be possible to distinguish the resulting processes because the observation of qubit q should not be influenced by the possible Hadamard operation on qubit p. From this, we conclude that the bisimilarity is too fine since it distinguishes processes that *should* be bisimilar, and hence that there is an observational property of the processes in Example 3.3 that is not taken into account.

From a theoretical point of view, we may consider whether there is a stronger relation that is preserved by parallel composition. Such a relation would necessarily have to distinguish the processes in Example 3.2. However, as with any physical theory, it is important to consider whether the theory accurately describes the reality. Whilst an arbitrary congruence may be theoretically interesting, our aim is to develop formal methods for *practical* quantum communication. We therefore consider the link between the observations described by CQP and the mathematical theory of quantum mechanics.

To analyse this link, let us consider the teleportation process from Figure 3.8. In particular, we are interested in quantifying the information that Bob has about the quantum state, directly before and after Alice makes her measurements. The quantum state before the measurement is

$$[\widetilde{r}pq_1q_2 \mapsto \frac{1}{2}|\phi_0\rangle(|000\rangle + |001\rangle + |110\rangle - |111\rangle) + \frac{1}{2}|\phi_1\rangle(|100\rangle + |101\rangle + |010\rangle - |011\rangle)] \ .$$

After the measurement there are four possibilities corresponding to the possible measurement outcomes. For example,

$$[\widetilde{r}pq_1q_2 \mapsto |\phi_0\rangle|000\rangle + |\phi_1\rangle|010\rangle] \ .$$

In these states, Bob owns the qubit q_1, and it is this qubit that determines the information he has about the quantum state. Specifically, we are considering Bob's ability to distinguish particular states; this ability is quantified by the reduced density matrix of q_1, ρ^{q_1}. This is the value that we would use in the bisimulation conditions, if Bob were to output his qubit.

Before the measurement we calculate the reduced density matrix of q_1 as

$$\rho^{q_1} = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) \ .$$

However, after the measurement, in one of the possible scenarios we calculate

$$\rho^{q_1} = \langle\phi_0|\phi_0\rangle|0\rangle\langle 0| + \langle\phi_0|\phi_1\rangle|0\rangle\langle 1| + \langle\phi_1|\phi_0\rangle|1\rangle\langle 0| + \langle\phi_1|\phi_1\rangle|1\rangle\langle 1| \ .$$

This implies that Bob is able to distinguish the quantum states occurring before and after Alice's measurements, using only his own qubit. In fact, this second calculation gives the same reduced density matrix as the initial qubit that Alice received, indicating that the state has been teleported. Given that Bob has not yet received the measurement results from Alice, this would imply that faster-than-light communication had occurred.

In this instance, the observation of the quantum state determined by the CQP semantics is not in agreement with quantum mechanics. The issue here is that, in this one of four possibilities, we have neglected to consider the other three outcomes that could have occurred. In order to take into account Bob's uncertainty about which particular outcome has been realised, we include all four possibilities, which results in the mixed state

$$\rho^{q_1} = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) \ .$$

This is identical to the reduced density matrix before Alice made the measurements, indicating that Bob learns nothing from Alice's action. This mixed state, from Bob's perspective, is exactly what the laws of quantum mechanics predicts.

This failure to consider all possibilities is a consequence of probabilistic branching. Each time a measurement occurs, one outcome is chosen probabilistically. This inadvertently affects all processes, as this example with Bob demonstrates. In the next chapter we discuss modifications to the semantics that respect the information known to individual processes.

3.6 Summary

We have undertaken an initial investigation into process equivalences for CQP processes. This investigation has centered around the properties of quantum teleportation. Driven by the perceived equivalence of teleportation and a quantum channel, the aim was to find a relation that identifies these respective processes.

In Section 3.1, we described the approach to modelling processes as components with external interactions in a similar style to Lalire [2006]. This contrasts to the closed system approach used in the original development of CQP [Gay and Nagarajan 2005]. In Section 3.1.2, we defined the operational semantics of CQP in terms of a labelled transition system. This complements the reduction relation of Gay and Nagarajan [2005], and provides the necessary rules to describe external interactions. Soundness of the type system, as proved by Gay and Nagarajan [2006], is re-stated and proved with respect to the labelled transition system in Section 3.1.3. This includes type preservation (Theorem 3.14) and the unique ownership of qubits (Theorem 3.15) which play a central role in Section 3.4.

A probabilistic branching bisimulation on configurations is defined in Section 3.2.1. We extend this to a relation on processes and prove that the result is an equivalence relation. In Section 3.3, we apply our probabilistic branching bisimilarity to quantum teleportation, proving that it is bisimilar to a direct quantum channel. We also prove that an alternative teleportation implementation and a qubit swap circuit are bisimilar to the quantum channel, and therefore to each other.

In Section 3.4, we consider preservation of bisimilarity with respect to the operators of CQP. We prove that bisimilarity is preserved by prefix and choice, but is not preserved by parallel composition. We prove, for the class of processes that are bisimilar to the direct quantum channel, that bisimilarity is also preserved by parallel composition.

We analyse the validity of the bisimilarity and the semantics in Section 3.5. In particular, we find that the description of the quantum state using density matrices, does not agree with the laws of quantum mechanics.

4

Congruence for Quantum Processes

The theory of computation has traditionally been studied almost entirely in the abstract, as a topic in pure mathematics. This is to miss the point of it. Computers are physical objects, and computations are physical processes. What computers can or cannot compute is determined by the laws of physics alone, and not by pure mathematics.

— David Deutsch

Chapter 3 presented an initial attempt at considering process equivalence within the CQP framework. The results highlighted a particularly important point; that the implementation and interpretation of probabilistic branching is not compatible with the theory of quantum mechanics.

The notion of behavioural equivalence is tightly integrated with the semantics of the language, since it is the latter which determines the capabilities of a process. The significant difference in the semantics presented in this chapter with respect to previous approaches is in the treatment of quantum measurements. In Chapter 3, and the previous work of Lalire [2006], a measurement results in a probabilistic distribution of configurations of which one is chosen by a probabilistic transition. Probabilistic branching is not used in qCCS, where instead the probabilistic information is retained by using distributions throughout [Feng et al. 2006; Ying et al. 2007, 2009]. The overall effect of these two approaches is that measurement is only considered at a single level – the global level – with respect to the processes. When considering observational equivalence this does not allow us to treat measurement as a local, or internal, action.

The aims of this chapter are twofold; to develop an accurate semantic model of quantum processes with external interactions, and to develop a realistic notion of observational equivalence. In Section 4.1, we discuss the observational properties of measurement and the role it plays in process calculus. We describe how *mixed states* should be used to describe the quantum state in specific scenarios, and we extend the notion of mixed states to configurations. In Section 4.2, we define the semantics for this radically new approach, and prove that the new transition relations preserve typing.

Process equivalence is considered in Section 4.3. We update the definition of probabilistic branching bisimilarity that was introduced in the previous chapter, adapting the quantum state conditions to suit the new transition rules. The preservation properties of this relation are considered in Section 4.3.1, and we prove that it is preserved by parallel composition. We then define full probabilistic branching bisimilarity and prove that it is a congruence.

In Section 4.4, we analyse the teleportation protocol with respect to the new semantic model and bisimilarity, and we are able to prove that it is congruent to its specification. We also prove the correctness of a superdense coding protocol.

4.1 Understanding Measurement

Mixed states were introduced in Section 2.1.5. In this section, we discuss mixed states in further detail, and their application to process calculus. In particular, we identify the relevance of mixed states to observers, while pure states should be considered for the process. In order to represent this distinction between observer and process, we extend the concept of mixed states to a similar notion based on configurations; we call this new concept a *mixed configuration*.

Mixed states provide the ability to represent *classical uncertainty* in the quantum state, such that can arise from a measurement in which the outcome is unknown. A mixed state is a probabilistic distribution, or ensemble, of pure states. Obtaining further information about the measurement outcome will alter the probability distribution, thereby resulting in a different mixed state. Indeed, such information may remove all classical uncertainty, in which case the result is one of the component pure states. It is important to note that although the mixed state may change, the component quantum states remain constant because there is no associated quantum operation.

Example 4.1. Consider a 2-qubit system in a state $|\psi\rangle = \alpha|0\rangle+\beta|1\rangle\otimes\gamma|0\rangle+\delta|1\rangle$. The possible measurement outcomes and respective probabilities are given in the following table.

State	Probability
$\|00\rangle$	$\|\alpha\|^2\|\gamma\|^2$
$\|01\rangle$	$\|\alpha\|^2\|\delta\|^2$
$\|10\rangle$	$\|\beta\|^2\|\gamma\|^2$
$\|11\rangle$	$\|\beta\|^2\|\delta\|^2$

If the specific outcome is unknown, then we can describe the quantum state as a probabilistic ensemble of these pure states. This mixed state is written as the density matrix

$$\rho = |\alpha|^2|\gamma|^2|00\rangle\langle 00| + |\alpha|^2|\delta|^2|01\rangle\langle 01| + |\beta|^2|\gamma|^2|10\rangle\langle 10| + |\beta|^2|\delta|^2|11\rangle\langle 11| \ .$$

If we discover that the first qubit is in state $|0\rangle$, then we can safely discount two of the possibilities, leaving only $|00\rangle$ and $|01\rangle$. This new mixed state is described by the density matrix

$$\rho' = |\gamma|^2|00\rangle\langle 00| + |\delta|^2|01\rangle\langle 01| \ .$$

Further information may indicate that the second qubit is in state $|1\rangle$, which allows us to isolate $|01\rangle$ as the only possible state; represented as a density matrix, this is the pure state $\rho'' = |01\rangle\langle 01|$. In each of these cases, the true quantum state has not changed, but our information has.

4.1.1 Measurement and Process Calculus

In the previous chapter, a measurement resulted in a process branching to one of the possible outcomes. This behaviour represents the assumption that, in a particular branch, there is no classical uncertainty. In other words, the measurement outcome is known for certain. Because branching occurs globally, then this knowledge must also be available to any context or observer.

One of the primary features of process calculus is the ability to treat internal behaviour abstractly. This principle does not fit perfectly with quantum mechanics, since quantum operations such as measurement affect the global quantum state and may therefore exhibit side effects. In particular, the combination of entanglement and measurement means that the effects of a quantum operation can be witnessed outside the target system. This is seen, for example, in the teleportation protocol, in which a measurement by Alice affects the state of Bob's qubit.

Let us return to the example of the teleportation protocol. During the execution of the protocol, Alice measures the two qubits in her possession. If, before this measurement, the state of the system was

$$|\psi\rangle = \frac{\alpha}{2}(|000\rangle + |101\rangle + |010\rangle - |111\rangle) + \frac{\beta}{\sqrt{2}}(|100\rangle + |110\rangle + |001\rangle - |011\rangle$$

where Alice has the first two qubits, then after the measurement there are four possible outcomes:

$$\begin{aligned}|\psi_0\rangle &= \alpha|000\rangle + \beta|001\rangle \\ |\psi_1\rangle &= \alpha|010\rangle - \beta|011\rangle \\ |\psi_2\rangle &= \beta|100\rangle + \alpha|101\rangle \\ |\psi_3\rangle &= \beta|110\rangle - \alpha|111\rangle\end{aligned}$$

each occurring with probability $\frac{1}{4}$. At this point, Alice knows which of the four states the system is in; we can say this because the values that she will send to Bob depend on the specific outcome. Meanwhile, Bob cannot be sure which of the four outcomes has occurred; he now has classical uncertainty. The reduced density matrix of Bob's qubit before the measurement is

$$\rho = \frac{1}{2}(|\alpha|^2|0\rangle\langle 0| + |\beta|^2|0\rangle\langle 0| + |\alpha|^2|1\rangle\langle 1| + |\beta|^2|1\rangle\langle 1|) \ .$$

After the measurement, taking into account the four possibilities and their respective probabilities, we have

$$\begin{aligned}\rho' = &\frac{1}{4}|\alpha|^2|0\rangle\langle 0| + \alpha^*\beta|0\rangle\langle 1| + \beta^*\alpha|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1| \\ &+ \frac{1}{4}|\alpha|^2|0\rangle\langle 0| - \alpha^*\beta|0\rangle\langle 1| - \beta^*\alpha|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1| \\ &+ \frac{1}{4}|\alpha|^2|0\rangle\langle 0| + \alpha^*\beta|1\rangle\langle 0| + \beta^*\alpha|0\rangle\langle 1| + |\beta|^2|1\rangle\langle 1| \\ &+ \frac{1}{4}|\alpha|^2|0\rangle\langle 0| + \alpha^*\beta|1\rangle\langle 0| - \beta^*\alpha|0\rangle\langle 1| + |\beta|^2|1\rangle\langle 1| \\ = &\frac{1}{2}(|\alpha|^2|0\rangle\langle 0| + |\beta|^2|0\rangle\langle 0| + |\alpha|^2|1\rangle\langle 1| + |\beta|^2|1\rangle\langle 1|) \ .\end{aligned}$$

This is the same as before the measurement, and represents the fact that Bob gains no information about the state of his qubit just by Alice measuring her qubits. The protocol proceeds by Alice sending the measurement results to Bob, which allows him to make the corrective operations in order to place his qubit in the state $\alpha|0\rangle + \beta|1\rangle$. It is only upon receiving the measurement results from Alice that Bob can be sure which of the four states his qubit is in. For example, if he receives the results $1, 0$ from Alice, then the reduced density matrix of his qubit will be

$$\rho'' = |\beta|^2|0\rangle\langle 0| + \beta\alpha^*|0\rangle\langle 1| + \alpha\beta^*|1\rangle\langle 0| + |\alpha|^2|1\rangle\langle 1| \ .$$

It is important to realise that if Bob could distinguish the state of his qubit before receiving the results from Alice, then it would be possible to complete the protocol

without the classical communication step. This would in fact mean that information was transmitted purely through measurement and entanglement.

In CQP, information about measurement results is stored in values; Bob's knowledge about Alice's measurements are described by the values that he receives. Significantly, these values are only sent to Bob, and therefore any other process, including an external observer, has no information about the measurement.

So, what happens when an external observer receives a measurement result? An external observer is essentially a *possible context*. If *Bob* were such a context for the process *Alice*, then the effect of receiving the values from *Alice* is to select one of four possible paths, because there is no longer any classical uncertainty. In process calculus, this choice corresponds to branching behaviour because all other options are discounted from this point. We can see from this that the branching does not occur at the point of measurement (as previously assumed), but instead at the transfer of values. Because we are considering an external observer, this transfer of values must correspond to an output action (as opposed to internal communication). We now consider how this information can be represented in CQP.

4.1.2 Mixed Configurations

It is not sufficient to include a mixed state in place of the state vector representation in a configuration, because this would not be able to represent the respective measurement values within the process. The solution we propose, is to use a probabilistic distribution over the quantum state *and* the measurement values; the result is what we shall call a *mixed configuration*. This is similar to the configuration distributions used in qCCS, however the significant difference is in the way we shall treat output.

The central idea behind this approach is that the result of a quantum measurement is known only locally to a process. This is illustrated in Example 4.2, in which a mixed configuration results from a measurement. Each component of the mixed configuration corresponds to the individual measurement outcomes, and the behaviour of each component will be dependent only on its respective outcome. The overall state of the system is still considered as a mixture of the two because the measurement outcome is not known outside the process.

Example 4.2.

$$([q \mapsto \alpha|0\rangle + \beta|1\rangle]; q; c![\mathsf{measure}\ q].P) \xrightarrow{\tau} \oplus_{i\in\{0,1\}}\ g_i\ ([q \mapsto |i\rangle]; q; \lambda x.c![x].P; i)$$

where $g_0 = |\alpha|^2$ and $g_1 = |\beta|^2$.

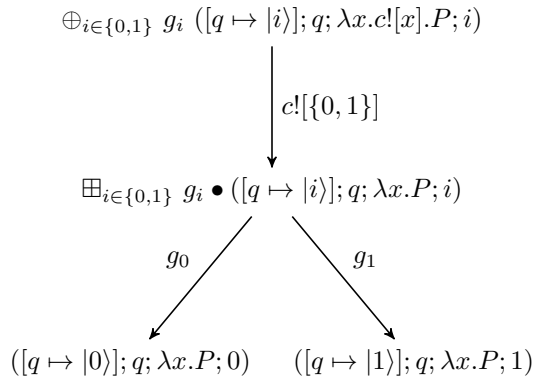
Here, the expression on the left of $\xrightarrow{\tau}$ is a *pure configuration*, consisting of a quantum state $[q \mapsto \alpha|0\rangle + \beta|1\rangle]$, a list of the qubits (just q) owned by the process, and

a process expression $c![\mathsf{measure}\ q].P$ which sends on channel c the result of measuring qubit q, then behaves as P. On the right of $\xrightarrow{\tau}$ we have a *mixed configuration* in which the $\oplus$ ranges over the possible outcomes of the measurement. The quantum state $[q \mapsto |i\rangle]$ corresponds to the measurement outcome. The expression $\lambda x.c![x].P$ is not a λ-calculus function, but represents the fact that the components of the mixed configuration have the same process structure and differ only in their values. The final term in the configuration, i, shows how the abstracted variable x should be instantiated in each component. So the mixed configuration expression is essentially an abbreviation of

$$g_0([q \mapsto |0\rangle]; q; c![0].P\{0/x\}) \oplus g_1([q \mapsto |1\rangle]; q; c![1].P\{1/x\}) \ .$$

If a measurement outcome is output then it becomes apparent to an observer which of the possible states the system is in. This is represented by probabilistic branching, after which we consider the system to be in one branch or the other — it is no longer a mixture of the two. In Example 4.3 the intermediate configuration is a probability distribution over pure configurations (in contrast to a mixed configuration; note the change from $\oplus$ to $\boxplus$) from which there are two possible probabilistic transitions.

Example 4.3.



Measurement outcomes may be communicated between processes without creating a probability distribution. In these cases an observer must still consider the system in a mixed configuration. In Example 4.4 there is no difference in process Q between the two components of the mixed configuration. However, after communication, the different possible values for x have been propagated to Q, so we include Q in the abstraction.

Example 4.4.

$$\oplus_{i\in\{0,1\}}\ g_i\ ([q \mapsto |i\rangle]; q; \lambda x.c![x].P \parallel c?[y].Q; i) \xrightarrow{\tau} \oplus_{i\in\{0,1\}}\ g_i\ ([q \mapsto |i\rangle]; q; \lambda x.(P \parallel Q\{x/y\}); i)$$

4.2 CQP with Mixed Configurations

4.2.1 Semantics

We now give more formal definitions of the operational semantics and the configurations used in it, before presenting a final example. A *pure configuration* has the same form as the configurations presented in Chapter 3, that is a triple $([\widetilde{q} \mapsto |\psi\rangle]; \omega; P)$. A *mixed configuration* is a weighted distribution over pure configurations.

Definition 4.1 (Mixed Configuration)**.** A *mixed configuration* is a weighted distribution, written

$$\oplus_{i\in I}\ g_i\ ([\widetilde{q} \mapsto |\psi_i\rangle]; \omega; \lambda\widetilde{x}.P; \widetilde{v}_i)$$

with weights g_i where $\sum_{i\in I} g_i = 1$ and for each $i \in I$, $0 < g_i \leq 1$ and $|\psi_i\rangle \in \mathcal{H}^{2^{|\widetilde{q}|}}$ and $|\widetilde{v}_i| = |\widetilde{x}|$.

The operator $\oplus$ (not to be confused with $\boxplus$ which represents probabilistic distributions) represents a distribution over the index set I with weights g_i. The process term is replaced by the expression $\lambda\widetilde{x}.P; \widetilde{v}_i$ which indicates that in each component the variables $\widetilde{x}$, appearing in P as placeholders, should be substituted for the values $\widetilde{v}_i$.

Although density operators do not appear explicitly in the definition, a mixed configuration induces a mixed state from the ensemble of pure states $|\psi_i\rangle$ within each component combined with the set of weights g_i. In correspondence with the observational properties of density operators, if two mixed configurations induce the same density operator, then their quantum states will be indistinguishable.

We will also make use of an *expanded* notation for mixed configurations which describes each pure component explicitly. The *expansion* $\mathcal{E}(s)$ of a mixed configuration s is defined as:

$$\mathcal{E}(\oplus_{i\in I}\ g_i\ ([\widetilde{q} \mapsto |\psi_i\rangle]; \omega; \lambda\widetilde{x}.P; \widetilde{v}_i)) = g_1([\widetilde{q} \mapsto |\psi_1\rangle]; \omega; P\{\widetilde{v}_1/\widetilde{x}\}) \oplus \cdots \oplus\ g_n\ ([\widetilde{q} \mapsto |\psi_n\rangle]; \omega; P\{\widetilde{v}_n/\widetilde{x}\})\ .$$

Although the expanded notation may be more convenient in many cases, the notation of Definition 4.1 is authoritative because it explicitly states the requirement for identical process terms (up to substitution of values) and qubit names in each component.

For example,

$$g_1([q \mapsto |\psi_1\rangle]; \omega; \{\mathsf{measure}\ q\}.\mathbf{0}) \oplus\ g_2\ ([q \mapsto |\psi_2\rangle]; q; \{q *\!\!= \mathsf{U}\}.\mathbf{0})$$

does not represent a valid mixed configuration. The expansion function $\mathcal{E}$ induces an equivalence on mixed configurations, that is, mixed configurations are identified upto equality of their expansion, thus $s \equiv t$ if and only if $\mathcal{E}(s) = \mathcal{E}(t)$.

We denote the set of mixed configurations by $\mathcal{M}$, and the set of pure configurations by $\mathcal{C}_p$. A pure configuration can be considered as a mixed configurations with a single component in the same way that pure states can be considered as mixed states with one component. "Mixed state" is often used as a blanket term when it is not necessarily known whether there is more than one component; we will use the term "mixed configuration" in a similar general manner, hence $\mathcal{C}_p \subset \mathcal{M}$.

Expression configurations are used for the evaluation of expressions (in a similar way to Chapter 3) and include both pure and mixed versions. *Pure expression configurations* are given by a tuple $([\widetilde{q} \mapsto |\psi\rangle]; \omega; e)$ as in Chapter 3. *Mixed expression configurations* are defined in a similar manner as mixed configurations; $\oplus_{i\in I}\ g_i\ ([\widetilde{q} \mapsto |\psi_i\rangle]; \omega; \lambda\widetilde{x}.e; \widetilde{v}_i)$ where the variables $\widetilde{x}$ are placeholders in the expression e. We denote the set of pure expression configurations by $\mathcal{C}_e$ and the set of mixed expression configurations by $\mathcal{M}_e$.

We retain the use of probabilistic configurations (probability distributions over configurations) in which the components are, in general, mixed configurations. In the abbreviated form, a probabilistic configuration is given by

$$\boxplus_{i\in I}\ p_i \bullet \oplus_{j\in J_i}\ g_{ij}\ ([\widetilde{q} \mapsto |\psi_{ij}\rangle]; \omega; \lambda\widetilde{x}.P; \widetilde{v}_{ij})$$

where each mixed configuration in the distribution may be over a different indexing set J_i. We assume $\sum_{i\in I} p_i = 1$ and for each i, $p_i > 0$. As each component is a mixed configuration then it must satisfy the requirement that for each i we have $\sum_{j\in J_i} g_{ij} = 1$.

Structural Congruence and Relabelling

We retain the rules of structural congruence from Figure 3.5 from the previous chapter, however we introduce stricter conditions on the permutation of quantum states and relabelling of qubit names. In particular, such operations must be limited to the *internal* quantum state.

Let π be a permutation on qubit names in ω and let Π be the corresponding permutation operator on the quantum state, then $([\widetilde{q} \mapsto |\psi\rangle]; \omega; P) \equiv ([\widetilde{q}\pi \mapsto \Pi|\psi\rangle]; \omega; P)$. Let f be a relabelling function where $f[q] = q$ for all $q \in \widetilde{q} \backslash \omega$, then $([\widetilde{q} \mapsto |\psi\rangle]; \omega; P) \equiv ([f[\widetilde{q}] \mapsto |\psi\rangle]; f[\omega]; P\{f[\widetilde{q}]/\widetilde{q}\})$ where f acts element-wise on lists.

$$([\widetilde{q} \mapsto |\psi\rangle]; \omega; u + v) \longrightarrow_v ([\widetilde{q} \mapsto |\psi\rangle]; \omega; \lambda x.x; w) \quad \text{where } w = u + v \qquad \text{(R-Plus)}$$

$$([q_0, \ldots, q_{n-1} \mapsto \alpha_0|\phi_0\rangle + \cdots + \alpha_{2^n-1}|\phi_{2^n-1}\rangle]; \omega; \mathsf{measure}\ q_0, \ldots, q_{r-1}) \longrightarrow_v \oplus_{0 \le m < 2^r}\ g_m\ ([q_0, \ldots, q_{n-1} \mapsto \tfrac{\alpha_{l_m}}{\sqrt{g_m}}|\phi_{l_m}\rangle + \cdots + \tfrac{\alpha_{u_m}}{\sqrt{g_m}}|\phi_{u_m}\rangle]; \omega; \lambda x.x; m)$$

(R-Measure)

$$\text{where } l_m = 2^{n-r}m, u_m = 2^{n-r}(m+1) - 1, g_m = |\alpha_{l_m}|^2 + \cdots + |\alpha_{u_m}|^2$$

$$([q_0, \ldots, q_{n-1} \mapsto |\phi\rangle]; \omega; q_0, \ldots, q_{r-1} *= U^m) \longrightarrow_v ([q_0, \ldots, q_{n-1} \mapsto (U^m \otimes I_{n-r})|\phi\rangle]; \omega; \mathsf{unit}; \cdot) \qquad \text{(R-Trans)}$$

$$\frac{\forall i \in I.([\widetilde{q} \mapsto |\psi_i\rangle]; \omega; e\{\widetilde{u}_i/\widetilde{y}\}) \longrightarrow_v \oplus_{j \in J_i}\ g_{ij}\ ([\widetilde{q} \mapsto |\psi_{ij}\rangle]; \omega; \lambda\widetilde{x}.e'\{\widetilde{u}_i/\widetilde{y}\}; \widetilde{v}_{ij})}{\oplus_{i \in I}\ h_i\ ([\widetilde{q} \mapsto |\psi_i\rangle]; \omega; \lambda\widetilde{y}.E[e]; \widetilde{u}_i) \longrightarrow_e \oplus_{\substack{i \in I \\ j \in J_i}}\ h_i g_{ij}\ ([\widetilde{q} \mapsto |\psi_{ij}\rangle]; \omega; \lambda\widetilde{y}\widetilde{x}.E[e']; \widetilde{u}_i, \widetilde{v}_{ij})} \qquad \text{(R-Context)}$$

Figure 4.1. Transition rules for values and expressions.

Mixed configurations are also considered congruent upto equality of their expansions, thus $s \equiv t$ if and only if $\mathcal{E}(s) = \mathcal{E}(t)$ (note that the operator $\oplus$ is commutative).

Expression Transition Rules

The transition relations $\longrightarrow_v \subseteq (\mathcal{C}_e \times \mathcal{M}_e)$ for evaluating values and $\longrightarrow_e \subseteq (\mathcal{M}_e \times \mathcal{M}_e)$ for evaluating expressions are defined by the rules in Figure 4.1. At first glance they are very similar to the rules in Figure 3.3, however the use of mixed expression configurations on the right hand side is key to the treatment of values in this new semantics. Instead of producing an expression configuration $(\sigma; \omega; w)$, R-Plus introduces a variable x as a placeholder for the value w. The importance of using a placeholder for a single value (there is only a single component) becomes apparent when we consider mixed expression configurations in R-Context because there may be a different value resulting from each component.

The result of a measurement is no longer a probabilistic configuration, but a mixed configuration in which each component corresponds to a specific outcome. Again, the variable x is introduced to maintain a constant expression term across all components, while the measurement value m is different for each component. Applying a unitary operator always results in the value unit, hence R-Trans does not need to introduce a new variable.

The rule R-Context has two primary purposes; it is used for the evaluation of expressions in an expression context E as in Chapter 3, and it also used for the evaluation of expressions in mixed configurations. The evaluation of a mixed expression configuration $\oplus_{i \in I}\ h_i\ (\sigma_i; \omega; \lambda\widetilde{y}.E[e]; \widetilde{u}_i)$ is determined by the evaluation of each component. For a given component, the pure expression configuration is obtained by substitution of the respective values; $(\sigma_i; \omega; E[e]\{\widetilde{u}_i/\widetilde{y}\})$. For this configuration we

$$([\widetilde{p}\widetilde{q}\widetilde{r} \mapsto |\psi\rangle]; \widetilde{p}, \widetilde{q}; c![\widetilde{v}, \widetilde{q}].P) \xrightarrow{c![\widetilde{v},\widetilde{q}]}_p ([\widetilde{p}\widetilde{q}\widetilde{r} \mapsto |\psi\rangle]; \widetilde{p}; P) \quad \text{(P-Out)}$$

$$([\widetilde{q} \mapsto |\psi\rangle]; \omega; c?[\widetilde{y}].P) \xrightarrow{c?[\widetilde{v},\widetilde{r}]}_p ([\widetilde{q} \mapsto |\psi\rangle]; \omega, \widetilde{r}; P\{\widetilde{v}, \widetilde{r}/\widetilde{y}\}) \quad \text{(P-In)}$$

$$\frac{([\widetilde{q} \mapsto |\psi\rangle]; \omega; P) \xrightarrow{\alpha}_p ([\widetilde{q} \mapsto |\psi\rangle]; \omega'; P')}{([\widetilde{q} \mapsto |\psi\rangle]; \omega; P \parallel Q) \xrightarrow{\alpha}_p ([\widetilde{q} \mapsto |\psi\rangle]; \omega'; P' \parallel Q)} \quad \text{(P-Par)}$$

$$\frac{([\widetilde{q} \mapsto |\psi\rangle]; \omega; P) \xrightarrow{\alpha}_p ([\widetilde{q} \mapsto |\psi\rangle]; \omega'; P')}{([\widetilde{q} \mapsto |\psi\rangle]; \omega; P + Q) \xrightarrow{\alpha}_p ([\widetilde{q} \mapsto |\psi\rangle]; \omega'; P')} \quad \text{(P-Sum)}$$

$$\frac{([\widetilde{q} \mapsto |\psi\rangle]; \omega; P) \xrightarrow{\alpha}_p ([\widetilde{q} \mapsto |\psi\rangle]; \omega; P')}{([\widetilde{q} \mapsto |\psi\rangle]; \omega; (\nu c)P) \xrightarrow{\alpha}_p ([\widetilde{q} \mapsto |\psi\rangle]; \omega; (\nu c)P')} \quad \text{if } \alpha \notin \{c?[\cdot], c![\cdot]\} \quad \text{(P-Res)}$$

Figure 4.2. Transition rules for pure process configurations.

isolate the context and consider the evaluation of $e\{\widetilde{u}_i/\widetilde{y}\}$. The resulting configuration may be a mixed expression configuration with new variables $\widetilde{x}$ introduced; specifically we end up with a term $\lambda\widetilde{x}.e'\{\widetilde{u}_i/\widetilde{y}\}; \widetilde{v}_{ij}$ where, due to the use of the substitution, e' is constant across each i. The results for each i are combined to give the final term $\lambda\widetilde{y}\widetilde{x}.E[e']; \widetilde{u}_i, \widetilde{v}_{ij}$ incorporating variables $\widetilde{x}$ and $\widetilde{y}$.

Pure Configuration Transition Rules

The transition relation $\xrightarrow{\alpha}_p \subseteq (\mathcal{C}_p \times \{?[], ![]\} \times \mathcal{C}_p)$ is defined by the rules in Figure 4.2. This relation is an addition to the semantics and defines input and output transitions for pure configurations. It is used in the hypotheses of L-Out and L-Com to determine the actions of the individual components in a mixed configurations. The inclusion of the choice (P-Sum), parallel (P-Par) and restriction (P-Res) rules are necessary to define input and output actions for arbitrary process constructions. We leave further discussion of this relation until the rules L-Out and L-Com have been formally introduced.

Mixed Configuration Transition Rules

The transition relation on mixed configurations, $\xrightarrow{\alpha} \subseteq (\mathcal{M} \times \{?[], ![], \tau\} \times \mathcal{M})$, is defined by the rules in Figure 4.3. The most obvious difference between these rules and the rules of Figure 3.4 is the change from pure to mixed configurations. Indeed, for many of the rules this change is straightforward; the general form that resulted in a probabilistic configuration ($s \xrightarrow{\alpha} \boxplus_i\ p_i \bullet s_i$) has been replaced by the introduction of more components ($\oplus_{i\in I}\ g_i\ s_i \xrightarrow{\alpha} \oplus_{i\in I, j\in J_i}\ g_i h_{ij}\ s_{ij}$).

The most interesting changes are to the rules L-In, L-Out and L-Com. Because the values associated with an input action are determined by the environment, this action is identical across all components in a mixed configuration. L-Par, L-Sum

$$\boxplus_j\ p_j \bullet (\oplus_i\ g_i\ (\sigma_i;\omega;P_i)) \overset{p_i}{\rightsquigarrow} \oplus_i\ g_i\ (\sigma_i;\omega;P_i) \qquad \text{(L-Prob)}$$

$$\oplus_i\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}.c?[\widetilde{y}].P;\widetilde{v}_i) \xrightarrow{c?[\widetilde{u},\widetilde{r}]} \oplus_i\ g_i\ (\sigma_i;\omega,\widetilde{r};\lambda\widetilde{x}.P\{\widetilde{u}/\widetilde{y}\};\widetilde{v}_i) \text{ where } |\widetilde{u}|+|\widetilde{r}|=|\widetilde{y}| \qquad \text{(L-In)}$$

$$\frac{\forall i \in I.([\widetilde{pq}\mapsto|\psi_i\rangle];\widetilde{p};P\{\widetilde{v}_i/\widetilde{x}\}) \xrightarrow{c![\widetilde{u}_i,\widetilde{r}]}_p ([\widetilde{pq}\mapsto|\psi_i\rangle];\widetilde{p}';P'\{\widetilde{v}_i/\widetilde{x}\})}{\oplus_{i\in I}\ g_i\ ([\widetilde{pq}\mapsto|\psi_i\rangle];\widetilde{p};\lambda\widetilde{x}.P;\widetilde{v}_i) \xrightarrow{c![U,\widetilde{r}]} \boxplus_{j\in J}\ p_j \bullet (\oplus_{i\in I_j}\ \frac{g_i}{p_j}\ ([\widetilde{p}'\widetilde{r}\widetilde{q}\mapsto\Pi|\psi_i\rangle];\widetilde{p}';\lambda\widetilde{x}.P';\widetilde{v}_i)) \qquad \text{(L-Out)}$$

where $U=\{\widetilde{u}_i \mid i\in I\}=\{\widetilde{u}_{k_j} \mid j\in J\}$ and $\forall j\in J, I_j=\{i|\widetilde{u}_i=\widetilde{u}_{k_j}\}, p_j=\sum_{i\in I_j} g_i$

and $\widetilde{r}\subseteq\widetilde{p},\widetilde{p}'=\widetilde{p}\setminus\widetilde{r},\Pi$ corresponds to the permutation $\pi:\widetilde{pq}\mapsto\widetilde{p}'\widetilde{r}\widetilde{q}$.

$$\frac{\begin{array}{c}\forall i\in I.(\sigma_i;\omega,\widetilde{r};P\{\widetilde{v}_i/\widetilde{x}\}) \xrightarrow{c![\widetilde{u}_i,\widetilde{r}]}_p (\sigma_i;\omega;P'\{\widetilde{v}_i/\widetilde{x}\}) \\ \forall i\in I.(\sigma_i;\omega;Q\{\widetilde{v}_i/\widetilde{x}\}) \xrightarrow{c?[\widetilde{u}_i,\widetilde{r}]}_p (\sigma_i;\omega,\widetilde{r};Q'\{\widetilde{v}_i/\widetilde{x}\})\end{array}}{\oplus_{i\in I}\ g_i\ (\sigma_i;\omega,\widetilde{r};\lambda\widetilde{x}.P \parallel Q;\widetilde{v}_i) \xrightarrow{\tau} \oplus_{i\in I}\ g_i\ (\sigma_i;\omega,\widetilde{r};\lambda\widetilde{x}.P' \parallel Q';\widetilde{v}_i)} \qquad \text{(L-Com)}$$

$$\frac{\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}.P;\widetilde{v}_i) \xrightarrow{\alpha} \oplus_{\substack{i\in I\\ j\in J_i}}\ g_i h_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{x}\widetilde{y}.P';\widetilde{v}_i\widetilde{w}_{ij})}{\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}.P \parallel Q;\widetilde{v}_i) \xrightarrow{\alpha} \oplus_{\substack{i\in I\\ j\in J_i}}\ g_i h_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{x}\widetilde{y}.P' \parallel Q;\widetilde{v}_i\widetilde{w}_{ij})} \qquad \text{(L-Par)}$$

$$\frac{\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}.P;\widetilde{v}_i) \xrightarrow{\alpha} \oplus_{\substack{i\in I\\ j\in J_i}}\ g_i h_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{x}\widetilde{y}.P';\widetilde{v}_i\widetilde{w}_{ij})}{\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}.P+Q;\widetilde{v}_i) \xrightarrow{\alpha} \oplus_{\substack{i\in I\\ j\in J_i}}\ g_i h_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{x}\widetilde{y}.P';\widetilde{v}_i\widetilde{w}_{ij})} \qquad \text{(L-Sum)}$$

$$\frac{\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}.P;\widetilde{v}_i) \xrightarrow{\alpha} \oplus_{\substack{i\in I\\ j\in J_i}}\ g_i h_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{x}\widetilde{y}.P';\widetilde{v}_i\widetilde{w}_{ij})}{\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}.(\nu c)P;\widetilde{v}_i) \xrightarrow{\alpha} \oplus_{\substack{i\in I\\ j\in J_i}}\ g_i h_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{x}\widetilde{y}.(\nu c)P';\widetilde{v}_i\widetilde{w}_{ij})} \qquad \text{(L-Res)}$$

if $\alpha\notin\{c?[\cdot],c![\cdot]\}$

$$\oplus_{i\in I}\ g_i\ ([\widetilde{q}\mapsto|\psi_i\rangle];\omega;\lambda\widetilde{x}.(\mathsf{qbit}\ y)P;\widetilde{v}_i) \xrightarrow{\tau} \oplus_{i\in I}\ g_i\ ([\widetilde{q},q\mapsto|\psi_i\rangle|0\rangle];\omega,q;\lambda\widetilde{x}.P\{q/y\};\widetilde{v}_i)$$

where q is fresh (L-Qbit)

$$\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}.\{u\}.P_i;\widetilde{v}_i) \xrightarrow{\tau} \oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}.P;\widetilde{v}_i) \qquad \text{(L-Act)}$$

$$\frac{\oplus_{i\in I}\ h_i\ (\sigma_i;\omega;\lambda\widetilde{y}.e;\widetilde{u}_i) \longrightarrow_e \oplus_{\substack{i\in I\\ j\in J_i}}\ h_i g_{ij}\ (\sigma_{ij};\omega;\lambda\widetilde{y}\widetilde{x}.e';\widetilde{u}_i\widetilde{v}_{ij})}{\oplus_{i\in I}\ h_i\ (\sigma_i;\omega;\lambda\widetilde{y}.F[e];\widetilde{u}_i) \xrightarrow{\tau} \oplus_{\substack{i\in I\\ j\in J_i}}\ h_i g_{ij}\ (\sigma_{ij};\omega;\lambda\widetilde{y}\widetilde{x}.F[e'];\widetilde{u}_i\widetilde{v}_{ij})} \qquad \text{(L-Expr)}$$

Figure 4.3. Transition rules for mixed process configurations.

and L-Res can then be used to define inputs on other process constructions in a mixed configuration. We note that if L-In was instead defined using $\longrightarrow_p$ for each component ($\forall i \in I$) then the derivations would not necessarily be unique, for example, P-Par could be used in place of L-Par to achieve parallel composition at the pure configuration level as opposed to the mixed configuration level.

The rule L-Out is the point at which mixed configurations are combined with probabilistic branching. Branching must occur when and only when there is information to distinguish the components. This information is represented by the classical values that are output, which may vary between the components. Some values may be the same, thereby requiring the relevant components to remain in a mixed configuration after the output. The purpose of L-Out is to distribute the components according to the different values, and to assign an action label that represents the combined action of *all* components. Each component has a pure transition $\stackrel{c![\widetilde{u}_i,\widetilde{r}]}{\longrightarrow_p}$ representing the channel and qubit names that are common to all components, and the values $\widetilde{u}_i$ that are specific to that component. The combined action label $c![U,\widetilde{r}]$ consists of these common elements and the set U of all the value tuples.

We now consider a detailed example to illustrate the new transition relations. This example focusses on the evaluation of mixed expressions, and the implementation of probabilistic branching for output actions.

Example 4.5. Let

$$s = ([q_1q_2 \mapsto (\alpha|0\rangle + \beta|1\rangle)(\gamma|0\rangle + \delta|1\rangle)]; q_1q_2; c![\mathsf{measure}\ q_1 + \mathsf{measure}\ q_2].P)$$

The first step of execution involves the evaluation of $\mathsf{measure}\ q_1$; the derivation is by L-Expr, R-Context and R-Measure:

$$\dfrac{\dfrac{([q_1q_2 \mapsto |\psi\rangle]; q_1q_2; \mathsf{measure}\ q_1) \longrightarrow_v \oplus_{i\in\{0,1\}}\ g_i\ ([q_1q_2 \mapsto |\psi_i\rangle]; q_1, q_2; \lambda x_1.x_1; i)}{([q_1q_2 \mapsto |\psi\rangle]; q_1, q_2; \mathsf{measure}\ q_1 + \mathsf{measure}\ q_2; \cdot) \longrightarrow_e \oplus_{i\in I}\ g_i\ ([q_1q_2 \mapsto |\psi_i\rangle]; q_1, q_2; \lambda x_1.x_1 + \mathsf{measure}\ q_2; i)}}{([q_1q_2 \mapsto |\psi\rangle]; q_1, q_2; c![\mathsf{measure}\ q_1 + \mathsf{measure}\ q_2].P; \cdot) \stackrel{\tau}{\longrightarrow} \oplus_{i\in I}\ g_i\ ([q_1q_2 \mapsto |\psi_i\rangle]; q_1, q_2; \lambda x_1.c![x_1 + \mathsf{measure}\ q_2].P; i)}$$

The next step involves the evaluation of measure q_2 by a similar derivation:

$$\frac{\begin{array}{l}([q_1q_2 \mapsto |\psi_0\rangle]; q_1, q_2; \mathsf{measure}\ q_2) \longrightarrow_v \oplus_{j\in\{0,1\}}\ h_{0j}\ ([q_1q_2 \mapsto |\psi_{0j}\rangle]; q_1, q_2; \lambda x_2.x_2; j) \\ ([q_1q_2 \mapsto |\psi_1\rangle]; q_1, q_2; \mathsf{measure}\ q_2) \longrightarrow_v \oplus_{j\in\{0,1\}}\ h_{1j}\ ([q_1q_2 \mapsto |\psi_{1j}\rangle]; q_1, q_2; \lambda x_2.x_2; j)\end{array}}{\begin{array}{l}\oplus_{i\in I}\ g_i\ ([q_1q_2 \mapsto |\psi_i\rangle]; q_1, q_2; \lambda x_1.x_1 + \mathsf{measure}\ q_2; i) \longrightarrow_e \\ \qquad \oplus_{\substack{i\in\{0,1\}\\ j\in\{0,1\}}}\ g_ih_{ij}\ ([q_1q_2 \mapsto |\psi_{ij}\rangle]; q_1, q_2; \lambda\widetilde{x}.x_1 + x_2; i, j)\end{array}}$$

$$\frac{\oplus_{i\in\{0,1\}}\ g_i\ ([q_1q_2 \mapsto |\psi_i\rangle]; q_1, q_2; \lambda x_1.c![x_1 + \mathsf{measure}\ q_2].P; i) \longrightarrow_e}{\oplus_{\substack{i\in\{0,1\}\\ j\in\{0,1\}}}\ g_ih_{ij}\ ([q_1q_2 \mapsto |\psi_{ij}\rangle]; q_1, q_2; \lambda\widetilde{x}.c![x_1 + x_2].P; i, j)}$$

The next step is the evaluation of the sum $x_1 + x_2$. R-Context uses the individual substitutions into each component of the configuration in order to evaluate all possible scenarios. The result is a variable x_3 with corresponding values $v_{ij} = i + j$:

$$\frac{\begin{array}{l}([q_1q_2 \mapsto |\phi_{00}\rangle]; q_1, q_2; 0 + 0) \longrightarrow_v ([q_1q_2 \mapsto |\phi_{00}\rangle]; q_1, q_2; \lambda x_3.x_3; 0) \\ ([q_1q_2 \mapsto |\phi_{01}\rangle]; q_1, q_2; 0 + 1) \longrightarrow_v ([q_1q_2 \mapsto |\phi_{01}\rangle]; q_1, q_2; \lambda x_3.x_3; 1) \\ ([q_1q_2 \mapsto |\phi_{10}\rangle]; q_1, q_2; 1 + 0) \longrightarrow_v ([q_1q_2 \mapsto |\phi_{10}\rangle]; q_1, q_2; \lambda x_3.x_3; 1) \\ ([q_1q_2 \mapsto |\phi_{11}\rangle]; q_1, q_2; 1 + 1) \longrightarrow_v ([q_1q_2 \mapsto |\phi_{11}\rangle]; q_1, q_2; \lambda x_3.x_3; 2)\end{array}}{\begin{array}{l}\oplus_{\substack{i\in\{0,1\}\\ j\in\{0,1\}}}\ g_ih_{ij}\ ([q_1q_2 \mapsto |\psi_{ij}\rangle]; q_1, q_2; \lambda\widetilde{x}.x_1 + x_2; i, j) \longrightarrow_e \\ \qquad \oplus_{\substack{i\in\{0,1\}\\ j\in\{0,1\}}}\ g_ih_{ij}\ ([q_1q_2 \mapsto |\psi_{ij}\rangle]; q_1, q_2; \lambda\widetilde{x}.x_3; i, j, v_{ij})\end{array}}$$

$$\frac{\oplus_{\substack{i\in\{0,1\}\\ j\in\{0,1\}}}\ g_ih_{ij}\ ([q_1q_2 \mapsto |\psi_{ij}\rangle]; q_1, q_2; \lambda\widetilde{x}.c![x_1 + x_2].P; i, j) \longrightarrow_e}{\oplus_{\substack{i\in\{0,1\}\\ j\in\{0,1\}}}\ g_ih_{ij}\ ([q_1q_2 \mapsto |\psi_{ij}\rangle]; q_1, q_2; \lambda\widetilde{x}.c![x_3].P; i, j, v_{ij})}$$

The final step is the output action which is derived by L-Out and P-Out:

$$\frac{\begin{array}{l}([q_1q_2 \mapsto |\psi_{00}\rangle]; q_1, q_2; c![0].P) \xrightarrow{c![0]}_p ([q_1q_2 \mapsto |\psi_{00}\rangle]; q_1, q_2; P) \\ ([q_1q_2 \mapsto |\psi_{01}\rangle]; q_1, q_2; c![1].P) \xrightarrow{c![1]}_p ([q_1q_2 \mapsto |\psi_{01}\rangle]; q_1, q_2; P) \\ ([q_1q_2 \mapsto |\psi_{10}\rangle]; q_1, q_2; c![1].P) \xrightarrow{c![1]}_p ([q_1q_2 \mapsto |\psi_{10}\rangle]; q_1, q_2; P) \\ ([q_1q_2 \mapsto |\psi_{11}\rangle]; q_1, q_2; c![2].P) \xrightarrow{c![2]}_p ([q_1q_2 \mapsto |\psi_{11}\rangle]; q_1, q_2; P)\end{array}}{\begin{array}{l}\oplus_{\substack{i\in\{0,1\}\\ j\in\{0,1\}}}\ g_ih_{ij}\ ([q_1q_2 \mapsto |\psi_{ij}\rangle]; q_1, q_2; \lambda\widetilde{x}.c![x_3].P; i, j, v_{ij}) \xrightarrow{c![U]} \\ \qquad \boxplus_{m\in U}\ p_m \bullet \oplus_{\substack{i\in I_m\\ j\in J_{im}}}\ g_ih_{ij}\ ([q_1q_2 \mapsto |\psi_{ij}\rangle]; q_1, q_2; \lambda\widetilde{x}.P; i, j, v_{ij})\end{array}}$$

where $U = \{0, 1, 2\}$ and $p_0 = g_0h_{00}$, $p_1 = g_1h_{10} + g_0h_{01}$, and $p_2 = g_1h_{11}$. Because the sum is 1 in two of the cases, this results in three branches. One of the branches remains a mixed configuration.

4.2.2 Type Soundness

In Section 3.1.3, we described the CQP type system and proved, with respect to the labelled transition system from 3.1.2, the two theorems from [Gay and Nagarajan 2006]; type preservation and the unique ownership of qubits. In this section, we re-address these proofs in the context of the semantics defined in Section 4.2.

We follow the same path as we work towards Theorem 4.4 (type preservation) and Theorem 4.5 (unique ownership of qubits). For the intermediate results that are independent of the semantics, we refer back to the presentation in Section 3.1.3. It remains to prove type preservation for each of the relations $\longrightarrow_v$, $\longrightarrow_e$, and $\xrightarrow{\alpha}_p$.

Lemma 4.1 (Type Preservation for $\longrightarrow_v$)**.** *If* $\Gamma;\Sigma \vdash e : T$ *and* $(\sigma;\omega;e) \longrightarrow_v \oplus_i g_i\ (\sigma_i;\omega';\lambda\widetilde{x}.e';\widetilde{v}_i)$ *and* $\Sigma \subseteq \omega$ *and* $\omega \subseteq dom(\sigma)$ *then* $\forall i.(dom(\sigma_i) = dom(\sigma))$ *and* $\omega' = \omega$ *and* $\Gamma, \widetilde{x}{:}\widetilde{T};\Sigma \vdash e' : T$ *and* $\forall i.(\Gamma;\emptyset \vdash \widetilde{v}_i : \widetilde{T})$.

Proof. By a straightforward case analysis of the derivation of the transition $(\sigma;\omega;e) \longrightarrow_v \oplus_i g_i\ (\sigma_i;\omega';\lambda\widetilde{x}.e;\widetilde{v}_i)$. □

Lemma 4.2 (Type Preservation for $\longrightarrow_e$)**.** *If* $\Gamma, \widetilde{y}{:}\widetilde{T};\Sigma \vdash e : T$ *and*

$$\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{y}.e;\widetilde{u}_i) \longrightarrow_e \oplus_{\substack{i\in I\\ j\in J_i}}\ g_i h_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{y}\widetilde{x}.e';\widetilde{u}_i,\widetilde{v}_{ij})$$

and $\Sigma \subseteq \omega$ *and* $\omega \subseteq dom(\sigma)$ *then* $\forall i.(dom(\sigma_i) = dom(\sigma))$ *and* $\omega' = \omega$ *and* $\Gamma, \widetilde{y}{:}T, \widetilde{x}{:}\widetilde{T}';\Sigma \vdash e' : T$ *and* $\forall i \in I, j \in J_i.(\Gamma;\emptyset \vdash \widetilde{v}_{ij} : \widetilde{T}')$.

Proof. The transition

$$\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{y}.e;\widetilde{u}_i) \longrightarrow_e \oplus_{\substack{i\in I\\ j\in J_i}}\ g_i h_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{y}\widetilde{x}.e';\widetilde{u}_i,\widetilde{v}_{ij})$$

is derived from R-Context, so for some E we have $e = E[f]$ and $e' = E[f']$ and for all $i \in I$,

$$(\sigma_i;\omega;f\{\widetilde{u}_i/\widetilde{y}\}) \longrightarrow_v \oplus_{j\in J_i}\ h_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{x}.f'\{\widetilde{u}_i/\widetilde{y}\};\widetilde{v}_{ij})\ .$$

From $\Gamma, \widetilde{y}{:}\widetilde{T};\Sigma \vdash E[f] : T$, Lemma 3.1 gives $\Gamma, \widetilde{y}{:}\widetilde{T};\Sigma \vdash f : U$ for some U. Lemma 3.11 gives $\forall i.(\Gamma;\Sigma \vdash f\{\widetilde{u}_i/\widetilde{y}\} : U)$ and Lemma 4.1 gives $\forall i.(\Gamma, \widetilde{x}{:}\widetilde{T}';\Sigma \vdash f'\{\widetilde{u}_i/\widetilde{y}\} : U)$ and $\forall i, j \in J_i.(\Gamma;\emptyset \vdash \widetilde{v}_{ij} : \widetilde{T}')$ and $\omega' = \omega$. Then Lemma 3.11 gives $\Gamma, \widetilde{y}{:}\widetilde{T}, \widetilde{x}{:}\widetilde{T}';\Sigma \vdash f' : U$. Because $\widetilde{x}$ are fresh, Lemma 3.7 gives $\Gamma, \widetilde{y}{:}\widetilde{T}, \widetilde{x}{:}\widetilde{T}';\Sigma \vdash f : U$ and then by Lemma 3.2 we arrive at $\Gamma, \widetilde{y}{:}\widetilde{T}, \widetilde{x}{:}\widetilde{T}';\Sigma \vdash E[f'] : T$. □

Lemma 4.3 (Type Preservation for $\xrightarrow{\alpha}_p$)**.** *If* $(\sigma;\omega;P) \xrightarrow{\alpha}_p (\sigma';\omega';P')$ *and* $\Gamma;\Sigma \vdash P$ *and* $\Sigma \subseteq \omega$ *and* $\omega \subseteq dom(\sigma)$ *then* $\sigma' = \sigma$ *and there exists* Σ' *such that* $\Sigma' \subseteq \omega'$ *and* $\omega' \subseteq dom(\sigma')$ *and* $\Gamma;\Sigma' \vdash P'$ *and if* $\Sigma \subseteq \Sigma'$ *then* $\Sigma' - \Sigma = \omega' - \omega$*, or if* $\Sigma' \subset \Sigma$ *then* $\Sigma - \Sigma' = \omega - \omega'$.

Proof. By induction on the derivation of $(\sigma;\omega;P) \xrightarrow{\alpha}_p (\sigma';\omega';P')$.

P-In: We have

$$(\sigma;\omega;c?[\widetilde{x}{:}\widetilde{T},\widetilde{y}{:}\widetilde{\mathsf{Qbit}}].P) \xrightarrow{c?[\widetilde{v},\widetilde{r}]}_p (\sigma;\omega';P\{\widetilde{v}\widetilde{r}/\widetilde{x}\widetilde{y}\})$$

where $\omega' = \omega,\widetilde{r}$. Then we have

$$\frac{\Gamma;\Sigma \vdash c : \widehat{}[\widetilde{T},\widetilde{\mathsf{Qbit}}] \quad \Gamma,\widetilde{x}{:}\widetilde{T},\widetilde{y}{:}\widetilde{\mathsf{Qbit}};\Sigma \vdash P}{\Gamma;\Sigma \vdash c?[\widetilde{x}{:}\widetilde{T},\widetilde{y}{:}\widetilde{\mathsf{Qbit}}].P}$$

where $\forall i.(T_i \neq \mathsf{Qbit})$. By Lemma 3.12 we have $\Gamma;\Sigma,\widetilde{r} \vdash P\{\widetilde{v}\widetilde{r}/\widetilde{x}\widetilde{y}\}$. Then $\Sigma,\widetilde{r} \subseteq \omega,\widetilde{r}$ and $\Sigma,\widetilde{r} - \Sigma = \omega,\widetilde{r} - \omega = \widetilde{r}$ and σ is constant. We must have $\widetilde{q} \in dom(\sigma)$, therefore $\omega,\widetilde{q} \subseteq dom(\sigma)$.

P-Out: We have $(\sigma;\omega;c![\widetilde{v},\widetilde{r}].P) \xrightarrow{c![\widetilde{v},\widetilde{r}]}_p (\sigma;\omega';P)$ where $\omega = \omega',\widetilde{r}$. Then we have

$$\frac{\Gamma;\Sigma' \vdash c : \widehat{}[\widetilde{T},\widetilde{\mathsf{Qbit}}] \quad \Gamma;\Sigma' \vdash P}{\Gamma;\Sigma \vdash c![\widetilde{v},\widetilde{r}].P}$$

where $\Sigma = \Sigma',\widetilde{r}$. Then $\Gamma;\Sigma' \vdash P$ and $\Sigma' \subseteq \omega'$ since $\Sigma',\widetilde{r} \subseteq \omega',\widetilde{r}$ and $\Sigma - \Sigma' = \omega - \omega' = \widetilde{r}$ and σ is unchanged.

P-Par: We have $(\sigma;\omega;P \parallel Q) \xrightarrow{\alpha}_p (\sigma;\omega'P' \parallel Q)$ with the hypothesis $(\sigma;\omega;P) \xrightarrow{\alpha}_p (\sigma;\omega';P')$ and type derivation

$$\frac{\Gamma_1;\Sigma_1 \vdash P \quad \Gamma_2;\Sigma_2 \vdash Q \quad \Gamma_1 + \Gamma_2 \text{ defined} \quad \Sigma_1 \cap \Sigma_2 = \emptyset}{\Gamma_1 + \Gamma_2;\Sigma_1 \cup \Sigma_2 \vdash P \parallel Q}$$

We are given that $\Sigma_1 \cup \Sigma_2 \subseteq \omega$, hence $\Sigma_1 \subseteq \omega$. Applying the inductive hypothesis gives Σ_1' such that $\Gamma_1;\Sigma_1' \vdash P'$ and $\Sigma_1' \subseteq \omega'$ and if $\Sigma_1 \subseteq \Sigma_1'$ then $\Sigma_1' - \Sigma_1 = \omega' - \omega$ and if $\Sigma_1' \subset \Sigma_1$ then $\Sigma_1 - \Sigma_1' = \omega - \omega'$.

If $\Sigma_1 \subseteq \Sigma_1'$ then $\Sigma_1' \cap \Sigma_2 = (\Sigma_1 \cup (\omega' - \omega)) \cap \Sigma_2 = (\Sigma_1' \cap \Sigma_2) \cup (\omega' - \omega \cap \Sigma_2) = \emptyset$ because $\Sigma_2 \subseteq \omega$. If $\Sigma_1' \subset \Sigma_1$ then $\Sigma_1' \cap \Sigma_2 \subseteq \Sigma_1 \cap \Sigma_2 = \emptyset$. Therefore (by IT-Par) we obtain $\Gamma_1 + \Gamma_2;\Sigma_1' \cup \Sigma_2 \vdash P' \parallel Q$.

P-Sum: We have

$$\frac{(\sigma;\omega;P) \xrightarrow{\alpha}_p (\sigma;\omega';P')}{(\sigma;\omega;P+Q) \xrightarrow{\alpha}_p (\sigma;\omega';P')}$$

and the type derivation

$$\frac{\Gamma;\Sigma \vdash P \qquad \Gamma;\Sigma \vdash Q}{\Gamma;\Sigma \vdash P + Q}$$

Applying the inductive hypothesis gives Σ' such that $\Sigma' \subseteq \omega'$ and $\omega' \subseteq dom(\sigma)$ and if $\Sigma \subseteq \Sigma'$ then $\Sigma' - \Sigma = \omega' - \omega$ and if $\Sigma' \subset \Sigma$ then $\Sigma - \Sigma' = \omega - \omega'$.

P-Res: The transition has the derivation

$$\frac{(\sigma;\omega;P) \xrightarrow{\alpha}_p (\sigma;\omega';P')}{(\sigma;\omega;(\nu c)P) \xrightarrow{\alpha}_p (\sigma;\omega';(\nu c)P')}$$

and the type derivation has the hypothesis $\Gamma, c{:}\widehat{\ }[\widetilde{T}]; \Sigma \vdash P$. Applying the inductive hypothesis gives $\Gamma, c{:}\widehat{\ }[\widetilde{T}]; \Sigma' \vdash P'$ and $\Sigma' \subseteq \omega'$ and $\omega' \subseteq dom(\sigma')$ and if $\Sigma \subseteq \Sigma'$ then $\Sigma' - \Sigma = \omega' - \omega$ and if $\Sigma' \subset \Sigma$ then $\Sigma - \Sigma' = \omega - \omega'$. Therefore (by IT-Res) $\Gamma, c{:}\widehat{\ }[\widetilde{T}]; \Sigma' \vdash (\nu c)P'$. □

We are now ready to prove type preservation for $\xrightarrow{\alpha}$.

Theorem 4.4 (Type Preservation for $\xrightarrow{\alpha}$)**.** *If* $\Gamma, \widetilde{x}{:}\widetilde{T}; \Sigma \vdash P$ *and* $\forall i.(\Gamma; \emptyset \vdash \widetilde{v}_i : \widetilde{T})$ *and* $\oplus_i\ g_i\ (\sigma_i; \omega; \lambda\widetilde{x}.P; \widetilde{v}_i) \xrightarrow{\alpha} \boxplus_m\ p_m \bullet \oplus_j\ h'_j\ (\sigma_{jm}; \omega'; \lambda\widetilde{y}.P; \widetilde{v}_{jm})$ *and* $\Sigma \subseteq \omega$ *and* $\omega \subseteq dom(\sigma)$ *then* $\forall m, j.(dom(\sigma) \subseteq dom(\sigma_{jm}))$ *and there exists* Σ' *such that* $\Sigma' \subseteq \omega'$ *and* $\forall m, j.(\omega' \subseteq dom(\sigma_{jm}))$ *and* $(\Gamma, \widetilde{y}{:}\widetilde{U}; \Sigma' \vdash P')$ *and* $\forall m, j.(\Gamma; \emptyset \vdash \widetilde{v}_{jm} : \widetilde{U})$ *and if* $\Sigma \subseteq \Sigma'$ *then* $\Sigma' - \Sigma = \omega' - \omega$*, or if* $\Sigma' \subset \Sigma$ *then* $\Sigma - \Sigma' = \omega - \omega'$.

Proof. By induction on the derivation of the transition $\oplus_i\ g_i\ (\sigma_i; \omega; \lambda\widetilde{x}.P; \widetilde{v}_i) \xrightarrow{\alpha} \boxplus_m\ p_m \bullet \oplus_j\ h_j\ (\sigma_{jm}; \omega'; \lambda\widetilde{y}.P'; \widetilde{v}_{jm})$.

L-Expr: For some evaluation context F we have $P = F[e]$ and $P' = F[e']$ and $\oplus_i\ g_i\ (\sigma_i; \omega; \lambda\widetilde{x}.e; \widetilde{v}_i) \longrightarrow_e \oplus_j\ h_j\ (\sigma'_j; \omega'; \lambda\widetilde{y}.e'; \widetilde{v}'_j)$. From the derivation $\mathcal{D}$ of $\Gamma; \Sigma \vdash F[e]$, Lemma 3.5 gives T such that $\mathcal{D}'$ is a subderivation of $\mathcal{D}$ concluding $\Gamma; \Sigma \vdash e : T$. Lemma 3.4 gives $\Gamma, \widetilde{y}{:}\widetilde{U}; \Sigma \vdash e' : T$ and $\forall j.(\Gamma; \emptyset \vdash \widetilde{v}'_j : \widetilde{U})$ and $\forall i, j.(dom(\sigma_i) = dom(\sigma'_j))$ and $\omega' = \omega$, and Lemma 3.6 gives $\Gamma; \Sigma \vdash F[e']$.

L-Out: We have the derivation

$$\frac{\forall i \in I.(\sigma_i; \omega; P\{\widetilde{v}_i/\widetilde{x}\}) \xrightarrow{c![\widetilde{u}_i, \widetilde{r}]}_p (\sigma'_i; \omega'; P'\{\widetilde{v}_i/\widetilde{x}\})}{\oplus_i\ g_i\ (\sigma_i; \omega; \lambda\widetilde{x}.P; \widetilde{v}_i) \xrightarrow{c![U, \widetilde{r}]} \boxplus_m\ p_m \bullet \oplus_{i \in I_m}\ g_i\ (\sigma'_i; \omega'; \lambda\widetilde{x}.P'; \widetilde{v}_i)}$$

where $\omega = \omega', \widetilde{r}$. Lemma 3.12 gives $\forall i.(\Gamma; \Sigma \vdash P\{\widetilde{v}_i/\widetilde{x}\})$ and Lemma 4.3 gives $\forall i.(\sigma'_i = \sigma_i)$ and Σ' such that $\Sigma' \subseteq \omega'$ and $\forall i.(\omega' \subseteq dom(\sigma'_i))$ and $\forall i.(\Gamma; \Sigma' \vdash P'\{\widetilde{v}_i/\widetilde{x}\})$ and if $\Sigma \subseteq \Sigma'$ then $\Sigma' - \Sigma = \omega' - \omega$ or if $\Sigma' \subset \Sigma$ then $\Sigma - \Sigma' = \omega - \omega'$. Then (by Lemma 3.12) we obtain $\Gamma, \widetilde{x}{:}\widetilde{T}; \Sigma' \vdash P'$.

L-In: We have the transition

$$\oplus_{i \in I}\ g_i\ (\sigma_i; \omega; \lambda\widetilde{z}.c?[\widetilde{x}{:}\widetilde{T}, \widetilde{y}{:}\widetilde{\mathsf{Qbit}}].P; \widetilde{v}_i) \xrightarrow{c?[\widetilde{u}, \widetilde{r}]} \oplus_{i \in I}\ g_i\ (\sigma_i; \omega'; \lambda\widetilde{z}.P\{\widetilde{u}\widetilde{r}/\widetilde{x}\widetilde{y}\}; \widetilde{v}_i)$$

where $\omega' = \omega, \widetilde{r}$ and we have

$$\frac{\Gamma, \widetilde{z}{:}\widetilde{T}; \Sigma \vdash c : \widehat{\ }[\widetilde{T}, \widetilde{\mathsf{Qbit}}] \quad \Gamma, \widetilde{z}{:}\widetilde{T}, \widetilde{x}{:}\widetilde{U}, \widetilde{y}{:}\widetilde{\mathsf{Qbit}}; \Sigma \vdash P}{\Gamma, \widetilde{z}{:}\widetilde{T}; \Sigma \vdash c?[\widetilde{x}{:}\widetilde{T}, \widetilde{y}{:}\widetilde{\mathsf{Qbit}}]\,.\,P}$$

where $\forall i.(T_i \neq \mathsf{Qbit})$. By Lemma 3.12 we have $\Gamma, \widetilde{z}{:}\widetilde{T}; \Sigma' \vdash P\{\widetilde{u}\widetilde{r}/\widetilde{x}\widetilde{y}\}$ where $\Sigma' = \Sigma, \widetilde{r}$. Then $\Sigma' \subseteq \omega'$ and $\Sigma' - \Sigma = \omega' - \omega = \widetilde{r}$ and σ is constant. $\widetilde{r}$ must exist, therefore $\omega' \subseteq dom(\sigma)$.

L-Com: We have

$$\frac{\begin{array}{c}\forall i.(\sigma_i; \omega; P\{\widetilde{v}_i/\widetilde{x}\}) \xrightarrow{c![\widetilde{u}_i,\widetilde{r}]}_p (\sigma_i; \omega, \widetilde{r}; P'\{\widetilde{v}_i/\widetilde{x}\}) \\ \forall i.(\sigma_i; \omega, \widetilde{r}; Q\{\widetilde{v}_i/\widetilde{x}\}) \xrightarrow{c?[\widetilde{u}_i,\widetilde{r}]}_p (\sigma_i; \omega; Q'\{\widetilde{v}_i/\widetilde{x}\})\end{array}}{\oplus_i\ g_i\ (\sigma_i; \omega, \widetilde{r}; P \parallel Q) \xrightarrow{\tau} \oplus_i\ g_i\ (\sigma_i; \omega, \widetilde{r}; P' \parallel Q')}$$

and the typing derivation

$$\frac{\Gamma_1; \Sigma_1 \vdash P \quad \Gamma_2; \Sigma_2 \vdash Q \quad \Gamma_1 + \Gamma_2 \text{ defined} \quad \Sigma_1 \cap \Sigma_2 = \emptyset}{\Gamma_1 + \Gamma_2; \Sigma_1 \cup \Sigma_2 \vdash P \parallel Q}.$$

Lemma 3.12 gives $\forall i.(\Gamma_1'; \Sigma_1 \vdash P\{\widetilde{v}_i/\widetilde{x}\})$ and $\forall i.(\Gamma_2'; \Sigma_2 \vdash Q\{\widetilde{v}_i/\widetilde{x}\})$ where $\Gamma_1 = \Gamma_1', \widetilde{x}{:}\widetilde{T}$ and $\Gamma_2 = \Gamma_2', \widetilde{x}{:}\widetilde{T}$. Applying the inductive hypothesis gives $\Sigma_1 \subseteq \omega$ and $\Sigma_2 \subseteq \omega, \widetilde{r}$ and Σ_1', Σ_2' such that $\Sigma_1' \subseteq \omega, \widetilde{r}$ and $\Sigma_2' \subseteq \widetilde{r}$ and $\Sigma_1' - \Sigma_1 = \widetilde{r}$ and $\Sigma_2 - \Sigma_2' = \widetilde{r}$ and $\forall i.(\omega, \widetilde{r} \subseteq dom(\sigma))$ and $\Gamma_1'; \Sigma_1' \vdash P'\{\widetilde{v}_i/\widetilde{x}\}$ and $\Gamma_2'; \Sigma_2' \vdash Q'\{\widetilde{v}_i/\widetilde{x}\}$. Then $\Sigma_1' \cup \Sigma_2' = \Sigma_1 \cup \Sigma_2$ and $\Sigma_1' \cap \Sigma_2' = (\Sigma_1 \cup \widetilde{q}) \cap (\Sigma_2 - \widetilde{q}) = (\Sigma_1 \cap (\Sigma_2 - \widetilde{q})) \cup (\widetilde{q} \cap (\Sigma_2 - \widetilde{q})) \subseteq \Sigma_1 \cap \Sigma_2 = \emptyset$. Then (by Lemma 3.12) we have $\Gamma_1; \Sigma_1' \vdash P'$ and $\Gamma_2; \Sigma_2' \vdash Q'$. Then by using IT-Par we obtain $\Gamma_1 + \Gamma_2; \Sigma_1' \cup \Sigma_2' \vdash P' \parallel Q'$.

L-Act: We have the transition $\oplus_i\ g_i\ (\sigma_i; \omega; \lambda\widetilde{x}.\{v\}.P; \widetilde{v}_i) \xrightarrow{\tau} \oplus_i\ g_i\ (\sigma_i; \omega; \lambda\widetilde{x}.P; \widetilde{v}_i)$. The typing derivation concluding $\Gamma, \widetilde{x}{:}\widetilde{T}; \Sigma \vdash \{v\}.P$ contains the required hypothesis $\Gamma, \widetilde{x}{:}\widetilde{T}; \Sigma \vdash P$. Γ, Σ and ω are unchanged.

L-Res: The transition has the derivation

$$\frac{\oplus_i\ g_i\ (\sigma_i; \omega; \lambda\widetilde{x}.P; \widetilde{v}_i) \xrightarrow{\alpha} \oplus_j\ h_j\ (\sigma_j'; \omega'; \lambda\widetilde{y}.P'; \widetilde{v}_j')}{\oplus_i\ g_i\ (\sigma_i; \omega; \lambda\widetilde{x}.(\nu c{:}\widehat{\ }[\widetilde{U}])P; \widetilde{v}_i) \xrightarrow{\alpha} \oplus_j\ h_j\ (\sigma_j'; \omega'; \lambda\widetilde{y}.(\nu c{:}\widehat{\ }[\widetilde{U}])P'; \widetilde{v}_j')}$$

The typing derivation has the hypothesis $\Gamma, c{:}\widehat{\ }[\widetilde{U}], \widetilde{x}{:}\widetilde{T}; \Sigma \vdash P$. Applying the inductive hypothesis gives $\Gamma, c{:}\widehat{\ }[\widetilde{U}], \widetilde{y}{:}\widetilde{T}'; \Sigma' \vdash P'$ where $\Sigma' \subseteq \omega'$ and $\forall j.(\omega' \subseteq dom(\sigma_j'))$ and $\forall i, j.(dom(\sigma_i) \subseteq dom(\sigma_j'))$ and if $\Sigma \subseteq \Sigma'$ then $\Sigma' - \Sigma = \omega' - \omega$ and if $\Sigma' \subseteq \Sigma$ then $\Sigma - \Sigma' = \omega - \omega'$. Therefore (by IT-Res) $\Gamma, \widetilde{y}{:}\widetilde{T}'; \Sigma' \vdash (\nu c{:}\widehat{\ }[\widetilde{U}])P'$.

L-Qbit: We have the transition

$$\oplus_i\ g_i\ (\sigma_i; \omega; \lambda\widetilde{x}.(\mathsf{qbit}\ y)P; \widetilde{v}_i) \xrightarrow{\tau} \oplus_i\ g_i\ (\sigma_i'; \omega, q; \lambda\widetilde{x}.P\{q/y\}; \widetilde{v}_i)$$

where q is fresh and the typing derivation

$$\frac{\Gamma, \widetilde{x}{:}\widetilde{T}, y{:}\mathsf{Qbit}; \Sigma \vdash P}{\Gamma, \widetilde{x}{:}\widetilde{T}; \Sigma \vdash (\mathsf{qbit}\ y)P}$$

Applying the inductive hypothesis and Lemma 3.12 gives the required judgement $\Gamma, \widetilde{x}{:}\widetilde{T}; \Sigma, q \vdash P\{q/y\}$. We have $\Sigma, \widetilde{q} \subseteq \omega, \widetilde{q}$ and $\Sigma, \widetilde{q} - \Sigma = \omega, \widetilde{q} - \omega$ and $\forall i.(dom(\sigma_i) \subseteq dom(\sigma'_i))$ and $\forall i.(\omega, \widetilde{q} \subseteq dom(\sigma'_i))$.

L-Sum: We have

$$\frac{\oplus_i\ g_i\ (\sigma_i; \omega; \lambda\widetilde{x}.P; \widetilde{v}_i) \xrightarrow{\alpha} \oplus_j\ h_j\ (\sigma'_j; \omega'; \lambda\widetilde{y}.P'; \widetilde{v}'_j)}{\oplus_i\ g_i\ (\sigma_i; \omega; \lambda\widetilde{x}.P + Q; \widetilde{v}_i) \xrightarrow{\alpha} \oplus_j\ h_j\ (\sigma'_j; \omega'; \lambda\widetilde{y}.P'; \widetilde{v}'_j)}$$

and

$$\frac{\Gamma, \widetilde{x}{:}\widetilde{T}; \Sigma \vdash P \qquad \Gamma, \widetilde{x}{:}\widetilde{T}; \Sigma \vdash Q}{\Gamma, \widetilde{x}{:}\widetilde{T}; \Sigma \vdash P + Q}$$

Applying the inductive hypothesis gives Σ' such that $\Sigma' \subseteq \omega'$ and $\forall i, j.(dom(\sigma_i) \subseteq dom(\sigma'_j))$ and $\forall j.(\omega' \subseteq dom(\sigma'_j))$ and $\Gamma, \widetilde{y}{:}\widetilde{T'}; \Sigma' \vdash P'$ and if $\Sigma \subseteq \Sigma'$ then $\Sigma' - \Sigma = \omega' - \omega$ and if $\Sigma' \subset \Sigma$ then $\Sigma - \Sigma' = \omega - \omega'$.

L-Par: The transition

$$\oplus_i\ g_i\ (\sigma_i; \omega; \lambda\widetilde{x}.P \parallel Q; \widetilde{v}_i) \xrightarrow{\alpha} \oplus_j\ h_j\ (\sigma'_j; \omega'; \lambda\widetilde{y}.P' \parallel Q; \widetilde{v}'_j)$$

has the hypothesis

$$\oplus_i\ g_i\ (\sigma_i; \omega; \lambda\widetilde{x}.P; \widetilde{v}_i) \xrightarrow{\alpha} \oplus_j\ h_j\ (\sigma'_j; \omega'; \lambda\widetilde{y}.P'; \widetilde{v}'_j)\ .$$

We have the typing derivation

$$\frac{\Gamma_1; \Sigma_1 \vdash P \quad \Gamma_2; \Sigma_2 \vdash Q \quad \Gamma_1 + \Gamma_2 \text{ defined} \quad \Sigma_1 \cap \Sigma_2 = \emptyset}{\Gamma_1 + \Gamma_2; \Sigma_1 \cup \Sigma_2 \vdash P \parallel Q}$$

We are given that $\Sigma_1 \cup \Sigma_2 \subseteq \omega$, hence $\Sigma_1 \subseteq \omega$. Applying the inductive hypothesis gives Σ'_1 such that $\forall i.(\Gamma_1; \Sigma'_1 \vdash P_i)$ and $\Sigma'_1 \subseteq \omega'$ and $\forall i, j.(dom(\sigma_i) \subseteq dom(\sigma'_j))$ and $\forall j.(\omega' \subseteq dom(\sigma_j))$ and if $\Sigma \subseteq \Sigma'$ then $\Sigma' - \Sigma = \omega' - \omega$ and if $\Sigma' \subset \Sigma$ then $\Sigma - \Sigma' = \omega - \omega'$.

If $\Sigma_1 \subseteq \Sigma'_1$ then $\Sigma'_1 \cap \Sigma_2 = (\Sigma_1 \cup (\omega' - \omega)) \cap \Sigma_2 = (\Sigma_1 \cap \Sigma_2) \cup (\omega' - \omega \cap \Sigma_2) = \emptyset$ because $\Sigma_2 \subseteq \omega$. If $\Sigma'_1 \subset \Sigma$ then $\Sigma'_1 \cap \Sigma_2 \subseteq \Sigma_1 \cap \Sigma_2 = \emptyset$. Therefore (by IT-Par) we obtain $\Gamma_1 + \Gamma_2; \Sigma'_1 \cup \Sigma_2 \vdash P' \parallel Q$. □

The following theorem is dependent only upon the typing rules and not the semantics, hence the proof in Chapter 3 is still relevant.

Theorem 4.5 (Unique Ownership of Qubits)**.** *If* $\Gamma; \Sigma \vdash P \parallel Q$ *then* $fq(P) \cap fq(Q) = \emptyset$.

4.3 Behavioural Equivalence

In the previous sections, we considered the observational effects of measurement and we defined a new operational semantics that respects these properties. In this section, we define a process equivalence with respect to the new semantics. The relation, probabilistic branching bisimilarity, is an adaptation of the equivalence in Chapter 3.

We consider the preservation properties of probabilistic branching bisimilarity in Section 4.3.1. In particular, by using mixed configurations in combination with probabilistic branching, we find that this relation is preserved by parallel composition. In Section 4.3.2, we extend the relation to define *full probabilistic branching bisimilarity* and prove that this is a congruence.

We will make extensive use of the density matrix formalism, in order to describe the quantum state of mixed configurations. Formally, we define the density matrix of a mixed configuration by the following inductive definition.

Definition 4.2 (Density Matrix of Configurations)**.** Let $\sigma_i = [\widetilde{p} \mapsto |\psi_i\rangle]$ and $\widetilde{q} \subseteq \widetilde{p}$ and $s_i = (\sigma_i; \omega; \lambda\widetilde{x}.P; \widetilde{v_i})$ and $s = \oplus_i\ g_i\ s_i$. Then

1. $\rho(\sigma_i) = |\psi_i\rangle\langle\psi_i|$,
2. $\rho^{\widetilde{q}}(\sigma_i) = \mathrm{tr}_{\widetilde{p}\setminus\widetilde{q}}(|\psi_i\rangle\langle\psi_i|)$,
3. $\rho(s_i) = \rho(\sigma_i)$,
4. $\rho^{\widetilde{q}}(s_i) = \rho^{\widetilde{q}}(\sigma_i)$,
5. $\rho(s) = \sum_i g_i \rho(s_i)$, and
6. $\rho^{\widetilde{q}}(s) = \sum_i g_i \rho^{\widetilde{q}}(s_i)$.

We also introduce the notation ρ_E to denote the reduced density matrix of the *environment* qubits. Formally, if $s = ([\widetilde{q} \mapsto |\psi\rangle]; \widetilde{p}; P)$ then $\rho_E(s) = \rho^{\widetilde{r}}(s)$ where $\widetilde{r} = \widetilde{q} \setminus \widetilde{p}$. The definition of ρ_E is extended to mixed configurations in the same manner as ρ.

We use the same notation to abbreviate transitions as in Chapter 3: Let $\xrightarrow{\tau}{}^{+}$ denote zero or one τ transitions; $\Longrightarrow$ denote zero or more τ transitions; and $\overset{\alpha}{\Longrightarrow}$ be equivalent to $\Longrightarrow\xrightarrow{\alpha}\Longrightarrow$.

We have defined the rules L-Out and L-Prob to maintain compatibility with the alternating probabilistic model. As a result, we use the same probabilistic function as before; let $\mu : \mathcal{S} \times \mathcal{S} \to [0, 1]$ be defined by

$$\mu(s,t) = \begin{cases} \pi, & \text{if } s \overset{\pi}{\rightsquigarrow} t \\ 1, & \text{if } s = t \text{ and } s \in \mathcal{S}_n \\ 0, & \text{otherwise.} \end{cases}$$

In Section 4.5, we discuss the possibility of using a non-alternating probabilistic model to reduce redundancy. Trčka and Georgievska [2008] also present a generalised version of this probabilistic function that is likely to be suitable.

We now define probabilistic branching bisimulation for mixed configurations. The primary difference between this and Definition 3.3 is the condition for output matching. In this case, we must take into account output as the source of probabilistic distributions. In particular, we require the individual components in the resulting probabilistic distributions to be related (condition IId) in addition to the overall configurations being related (condition IIa). We also require the respective probabilities to match (condition IIb).

Definition 4.3 (Probabilistic Branching Bisimulation)**.** An equivalence relation $\mathcal{R}$ is a *probabilistic branching bisimulation* on configurations if whenever $(s,t) \in \mathcal{R}$ the following conditions are satisfied.

I. If $s \in \mathcal{S}_n$ and $s \xrightarrow{\tau} s'$ then there exists t', t'' such that $t \Longrightarrow t' \xrightarrow{\tau}{}^{+} t''$ where

   a) $(s,t') \in \mathcal{R}$, and

   b) $(s',t'') \in \mathcal{R}$.

II. If $s \xrightarrow{c![V,\widetilde{q}_1]} s'$ where $s' = \boxplus_{j\in\{1\ldots m\}}\ p_j \bullet s'_j$ and $V = \{\widetilde{v}_1, \ldots, \widetilde{v}_m\}$ then $\exists t', t''$ such that $t \Longrightarrow t' \xrightarrow{c![V,\widetilde{q}_2]} t''$ where

   a) $(s,t') \in \mathcal{R}$,

   b) $t'' = \boxplus_{j\in\{1\ldots m\}}\ p_j \bullet t''_j$,

   c) for each $j \in \{1,\ldots,m\}$, $\rho_E(s'_j) = \rho_E(t''_j)$.

   d) for each $j \in \{1,\ldots,m\}$, $(s'_j, t''_j) \in \mathcal{R}$.

III. If $s \xrightarrow{c?[\widetilde{v}]} s'$ then $\exists t', t''$ such that $t \Longrightarrow t' \xrightarrow{c?[\widetilde{v}]} t''$ where

   a) $(s,t') \in \mathcal{R}$,

   b) $(s',t'') \in \mathcal{R}$,

IV. If $s \in \mathcal{S}_p$ then $\mu(s,D) = \mu(t,D)$ for all classes $D \in \mathcal{S}/\mathcal{R}$.

In condition II we require that the distinct set of values V must match, but the qubit names ($\widetilde{q}_1$ and $\widetilde{q}_2$) need not be identical. Instead, for each pair of probabilistic components s'_j, t''_j, the respective reduced density matrices of the output qubits, combined with qubits $\widetilde{r}$ from the environment, must be identical ($\rho_E(s'_j) = \rho_E(t''_j)$).

Condition IV provides the matching on probabilistic configurations following the approach of Trčka and Georgievska [2008]. In this relation, a probabilistic configuration which necessarily evolves from an output will satisfy IV if the prior configuration

satisfies IId. It is necessary to include the latter condition to ensure that the probabilities are paired with their respective configurations.

Naturally this leads on to the following definition of bisimilarity on configurations.

Definition 4.4 (Probabilistic Branching Bisimilarity)**.** Let s and t be configurations. Then s and t are *probabilistic branching bisimilar*, denoted $s \leftrightarroweq t$ if and only if there exists a probabilistic branching bisimulation $\mathcal{R}$ such that $(s,t) \in \mathcal{R}$.

Following the same approach as in Chapter 3, we define equivalence of processes in terms of equivalence of configurations, by requiring independence from the quantum state. Again, we assume the qubit list is empty because we are considering processes prior to execution. In Section 4.3.2, we extend this to a relation that recognises executing processes with non-empty qubit lists.

Definition 4.5 (Probabilistic Branching Bisimilarity of Processes)**.** Let P and Q be processes. P and Q are *probabilistic branching bisimilar*, denoted $P \leftrightarroweq Q$, if and only if for all σ, $(\sigma; \emptyset; P) \leftrightarroweq (\sigma; \emptyset; Q)$.

It does not follow directly from the definition that probabilistic branching bisimilarity of processes is an equivalence relation. This property is dependent on the definition of bisimulation and not the semantics, hence the proof is similar to the corresponding result in Chapter 3.

Lemma 4.6. *Probabilistic branching bisimilarity of processes is an equivalence relation.*

Proof. Follows the same reasoning as Lemma 3.17 with minor modifications to the output case. □

4.3.1 Preservation Properties

We now consider the preservation properties of bisimilarity on processes. The first main result that we begin working towards is that bisimilarity is a *non-input, non-qubit congruence* (Theorem 4.17). The key to this result is that the bisimilarity is preserved by parallel composition (Theorem 4.14); with the exception of the recent, independent work by Feng et al. [2011], this is the property that previous quantum process equivalences have failed to possess.

Before continuing, we formally define *contexts* and *congruence*, and their *non-input,non-qubit* variants. The reason for considering variants without input and qubit declaration prefixes, is that substitution must also be considered when these are included. In Section 4.3.2, we define full probabilistic branching bisimilarity, which will also consider invariance under substitution.

Definition 4.6 (Context)**.** A *context* C is a process with a non-degenerate occurrence of $\mathbf{0}$ replaced by a hole, $[\cdot]$. Formally,

$$C ::= \ [] \ \mid \ (C \parallel P) \ \mid \ \alpha.C + P \ \mid \ \alpha.C \ \mid \ (\nu x{:}\widehat{\ }[T])C$$

for $\alpha \in \{e?[\tilde{x} : \tilde{T}], e![\tilde{e}], \{e\}, (\mathsf{qbit}\ x)\}$.

Definition 4.7 (Congruence)**.** An equivalence relation $\mathcal{R}$ on processes is a *congruence* if $(C[P], C[Q]) \in \mathcal{R}$ whenever $(P, Q) \in \mathcal{R}$ and C is a context.

Definition 4.8 (Non-input, non-qubit context)**.** A *non-input, non-qubit context* is a context in which the hole does not appear under an input or qubit declaration.

Definition 4.9 (Non-input, non-qubit congruence)**.** An equivalence relation $\mathcal{R}$ on processes is a *non-input, non-qubit congruence* if $(C[P], C[Q]) \in \mathcal{R}$ whenever $(P, Q) \in \mathcal{R}$ and C is a non-input, non-qubit context.

This first lemma provides a general form for representing mixed configurations related by internal transitions without introducing excessive indexing sets. As such, it's primary purpose is to simplify notation in the following proofs.

Lemma 4.7 (General form of internal transitions)**.** *If* $s = \oplus_{\substack{i\in I_j \\ j\in J}} g_{ij}\ (\sigma_{ij}; \widetilde{q}; \lambda\widetilde{x}.P; \widetilde{v}_{ij})$ *and* $s \Longrightarrow s'$ *then there exist sets* I'_j *such that* $s' = \oplus_{\substack{i\in I'_j \\ j\in J}} g'_{ij}\ (\sigma'_{ij}; \widetilde{q}'; \lambda\widetilde{x}'.P'; \widetilde{v}'_{ij})$.

Proof. By induction on the length of the sequence of τ-transitions. The inductive step is proved by a straightforward induction on the derivation of the transition. □

The following 3 lemmas prove that the state of qubits that are not owned by a particular process is unaffected by any transitions of that process. For example, in the proof of Lemma 4.8 we see that measurement of a set of qubits does not affect the reduced density matrix of other qubits.

Lemma 4.8 (External state independence for $\longrightarrow_v$)**.** *If* $\Gamma; \widetilde{q}_1 \vdash e : T$ *and* $s \longrightarrow_v s'$ *where* $s = ([\widetilde{q}_1\widetilde{q}_2\widetilde{q}_3 \mapsto |\psi\rangle]; \widetilde{q}_1, \widetilde{q}_2; e)$ *then* $\rho^{\widetilde{q}_2\widetilde{q}_3}(s) = \rho^{\widetilde{q}_2\widetilde{q}_3}(s')$.

Proof. By case analysis.

R-Plus: The quantum state and distribution are unchanged.

R-Trans: We have

$$s' = ([\widetilde{q}_1\widetilde{q}_2\widetilde{q}_3 \mapsto |\psi'\rangle]; \widetilde{q}_1, \widetilde{q}_2; \mathsf{unit})$$

where $|\psi'\rangle = (U^m \otimes I)|\psi\rangle$. We can write $|\psi\rangle = \sum_i |\psi_i\rangle_1|\phi_i\rangle_{23}$, then $|\psi'\rangle = \sum_i U^m|\psi_i\rangle_1 I|\phi_i\rangle_{23}$. Now

$$\begin{aligned}\rho^{\widetilde{q}_2\widetilde{q}_3}(s') &= \sum_{jk}\langle\psi'_j|\psi'_k\rangle_1|\phi_j\rangle\langle\phi_k|_{23} \\ &= \sum_{jk}\langle\psi_j|(U^m)^*U^m|\psi_k\rangle_1 I|\phi_j\rangle\langle\phi_k|_{23}I^* \\ &= \sum_{jk}\langle\psi_j|\psi_k\rangle_1|\phi_j\rangle\langle\phi_k|_{23} \\ &= \rho^{\widetilde{q}_2\widetilde{q}_3}(s)\ .\end{aligned}$$

R-Measure: We have the transition

$$\begin{aligned}&([\widetilde{q}_1\widetilde{q}_2\widetilde{q}_3 \mapsto |\psi\rangle]; \widetilde{q}_1, \widetilde{q}_2; \mathsf{measure}\ p_1 \dots p_{r-1}) \\ &\quad \longrightarrow_v \oplus_{0\le m<2^r}\ g_m\ ([\widetilde{q}_1\widetilde{q}_2\widetilde{q}_3 \mapsto |\psi_m\rangle]; \widetilde{q}_1, \widetilde{q}_2; \lambda x.x; m)\end{aligned}$$

where $\widetilde{q}_1 = p_0, \dots, p_{n-1}$ and $r \le n$ and

$$|\psi\rangle = \sum_{m=0}^{2^r-1}|\psi_m\rangle \text{ and } |\psi_m\rangle = \sum_{i=0}^{2^{N-r}-1}\frac{\alpha_{im}}{\sqrt{g_m}}|\phi_m\rangle|\phi'_i\rangle$$

where $N = |\widetilde{q}_1\widetilde{q}_2\widetilde{q}_3|$ and $\{|\phi_m\rangle\}$ is an orthonormal basis for qubits $p_0, \dots, p_{r-1}$ and $\{|\phi'_i\rangle\}$ is an orthonormal basis for qubits $p_r, \dots, p_{n-1}, \widetilde{q}_2, \widetilde{q}_3$. Then

$$\begin{aligned}\mathrm{tr}_{p_0\dots p_{r-1}}(|\psi\rangle) &= \sum_{\substack{m=0\\k=0}}^{2^r-1}\sum_{\substack{i=0\\j=0}}^{2^{N-r}-1}\alpha_{im}\alpha^*_{jk}\langle\phi_m|\phi_k\rangle|\phi'_i\rangle\langle\phi'_j| \\ &= \sum_{m=0}^{2^r-1}\sum_{\substack{i=0\\j=0}}^{2^{N-r}-1}\alpha_{im}\alpha^*_{jk}|\phi'_i\rangle\langle\phi'_j|\end{aligned} \tag{4.1}$$

since $\langle\phi_m|\phi_k\rangle = 1$ if $m = k$ and 0 otherwise. Now

$$\begin{aligned}\mathrm{tr}_{p_0\dots p_{r-1}}(|\psi_m\rangle) &= \sum_{\substack{i=0\\j=0}}^{2^{N-r}-1}\frac{\alpha_{im}\alpha^*_{jk}}{g_m}\langle\phi_m|\phi_m\rangle|\phi'_i\rangle\langle\phi'_j| \\ &= \frac{1}{g_m}\sum_{\substack{i=0\\j=0}}^{2^{N-r}-1}\alpha_{im}\alpha^*_{jk}|\phi'_i\rangle\langle\phi'_j|\ .\end{aligned} \tag{4.2}$$

Let $s' = \oplus_{0 \leq m < 2^r}\ g_m\ s'_m$, then we have

$$\begin{aligned}
\rho^{\widetilde{q}_2\widetilde{q}_3}(s') &= \sum_{m=0}^{2^r-1} g_m \rho^{\widetilde{q}_2\widetilde{q}_3}(s'_m) \\
&= \sum_{m=0}^{2^r-1} g_m \mathrm{tr}_{p_r \ldots p_{n-1}}(\mathrm{tr}_{p_1 \ldots p_{r-1}}(|\psi_m\rangle)) \\
&= \sum_{m=0}^{2^r-1} g_m \mathrm{tr}_{p_r \ldots p_{n-1}} \left(\frac{1}{g_m} \sum_{\substack{i=0 \\ j=0}}^{2^{N-r}-1} \alpha_{im}\alpha^*_{jk} |\phi'_i\rangle\langle\phi'_j| \right) \qquad \text{by (4.2)} \\
&= \mathrm{tr}_{p_r \ldots p_{n-1}} \left(\sum_{m=0}^{2^r-1} \sum_{\substack{i=0 \\ j=0}}^{2^{N-r}-1} \alpha_{im}\alpha^*_{jk} |\phi'_i\rangle\langle\phi'_j| \right) \\
&= \mathrm{tr}_{p_r \ldots p_{n-1}}(\mathrm{tr}_{p_0 \ldots p_{r-1}}(|\psi\rangle)) \qquad \text{by (4.1)} \\
&= \rho^{\widetilde{q}_2\widetilde{q}_3}(s)\ .
\end{aligned}$$

□

Lemma 4.9 (External state independence for $\longrightarrow_e$)**.** *If* $\Gamma; \widetilde{q}_1 \vdash e : T$ *and* $s \longrightarrow_e s'$ *where* $s = \oplus_{i \in I}\ g_i\ ([\widetilde{q}_1\widetilde{q}_2\widetilde{q}_3 \mapsto |\psi_i\rangle]; \widetilde{q}_1, \widetilde{q}_2; \lambda\widetilde{x}.e; \widetilde{v}_i)$ *then* $\rho^{\widetilde{q}_2\widetilde{q}_3}(s) = \rho^{\widetilde{q}_2\widetilde{q}_3}(s')$.

Proof. The transition $s \longrightarrow_e s'$ is derived by R-Context with a hypothesis

$$\forall i \in I . s_i \longrightarrow_v s'_i\ .$$

where $([\widetilde{q}_1\widetilde{q}_2\widetilde{q}_3 \mapsto |\psi_i\rangle]; \widetilde{q}_1, \widetilde{q}_2; e\{\widetilde{v}_i/\widetilde{x}\})$. For each $i \in I$ we have $\rho^{\widetilde{q}_2\widetilde{q}_3}(s_i) = \rho^{\widetilde{q}_2\widetilde{q}_3}(s'_i)$ by Lemma 4.8. From Definition 4.2 we have $\rho^{\widetilde{q}_2\widetilde{q}_3}(s) = \sum_{i \in I} \rho^{\widetilde{q}_2\widetilde{q}_3}(s_i)$ and $\rho^{\widetilde{q}_2\widetilde{q}_3}(s') = \sum_{i \in I} \rho^{\widetilde{q}_2\widetilde{q}_3}(s'_i)$, therefore we arrive at the equality $\rho^{\widetilde{q}_2\widetilde{q}_3}(s) = \rho^{\widetilde{q}_2\widetilde{q}_3}(s')$. □

Lemma 4.10 (External state independence for $\xrightarrow{\tau}$)**.** *If* $\Gamma; \widetilde{q}_1 \vdash P$ *and* $s \xrightarrow{\tau} s'$ *where* $s = \oplus_{i \in I}\ g_i\ ([\widetilde{q}_1\widetilde{q}_2\widetilde{q}_3 \mapsto |\psi_i\rangle]; \widetilde{q}_1, \widetilde{q}_2; \lambda\widetilde{x}.P; \widetilde{v}_i)$ *then* $\rho^{\widetilde{q}_2\widetilde{q}_3}(s) = \rho^{\widetilde{q}_2\widetilde{q}_3}(s')$.

Proof. By induction on the derivation of the transition $s \xrightarrow{\tau} s'$. Cases L-Par, L-Sum and L-Res are straightforward applications of the inductive hypothesis. The quantum state and distribution are unchanged for L-Com and L-Act, therefore these cases are trivial.

L-Qbit: We have a transition $\oplus_{i \in I}\ g_i\ s_i \xrightarrow{\tau} \oplus_{i \in I}\ g_i\ s'_i$ where for each $i \in I$, $\rho(s'_i) = \rho(s_i) \otimes |0\rangle\langle 0|$. Therefore $\rho^{\widetilde{q}_2\widetilde{q}_3}(s'_i) = \rho^{\widetilde{q}_2\widetilde{q}_3}(s_i) \otimes \langle 0|0\rangle = \rho^{\widetilde{q}_2\widetilde{q}_3}(s_i)$. Then we have $\rho^{\widetilde{q}_2\widetilde{q}_3}(s') = \rho^{\widetilde{q}_2\widetilde{q}_3}(s)$.

L-Expr: We have $P = F[e]$ and $P' = F[e']$ for some process context F and the

hypothesis

$$t = \oplus_{i\in I}\ g_i\ ([\widetilde{q}_1\widetilde{q}_2\widetilde{q}_3 \mapsto |\psi_i\rangle]; \widetilde{q}_1, \widetilde{q}_2; \lambda\widetilde{x}.e; \widetilde{v}_i) \longrightarrow_e \\ \oplus_{\substack{i\in I\\ j\in J_i}}\ g_i h_{ij}\ ([\widetilde{q}_1\widetilde{q}_2\widetilde{q}_3 \mapsto |\psi_{ij}\rangle]; \widetilde{q}_1, \widetilde{q}_2; \lambda\widetilde{x}\widetilde{y}.e'; \widetilde{v}_i, \widetilde{v}_{ij}) = t'\ .$$

By Lemma 4.9 we have $\rho^{\widetilde{q}_2\widetilde{q}_3}(t) = \rho^{\widetilde{q}_2\widetilde{q}_3}(t')$. It follows from the definition that $\rho^{\widetilde{q}_2\widetilde{q}_3}(s) = \rho^{\widetilde{q}_2\widetilde{q}_3}(t)$ and $\rho^{\widetilde{q}_2\widetilde{q}_3}(s') = \rho^{\widetilde{q}_2\widetilde{q}_3}(t')$, hence we get $\rho^{\widetilde{q}_2\widetilde{q}_3}(s) = \rho^{\widetilde{q}_2\widetilde{q}_3}(s')$. □

The next lemma proves that the action of a context on the quantum state is independent of the quantum subsystem owned by a process.

Lemma 4.11 (Independence of context transitions)**.** *Assume that* $\Gamma; \widetilde{q}_R \vdash R$*. Let* s *and* t *be configurations where*

$$s = \oplus_{i\in I}\ g_i\ ([\widetilde{q}_P\widetilde{q}_R\widetilde{q}_E \mapsto |\psi_i\rangle]; \widetilde{q}_P, \widetilde{q}_R; \lambda\widetilde{x}.R; \widetilde{v}_R) \\ t = \oplus_{j\in J}\ h_j\ ([\widetilde{q}_Q\widetilde{q}_R\widetilde{q}_E \mapsto |\phi_j\rangle]; \widetilde{q}_Q, \widetilde{q}_R; \lambda\widetilde{x}.R; \widetilde{v}_R)$$

If $\rho^{\widetilde{q}_R\widetilde{q}_E}(s) = \rho^{\widetilde{q}_R\widetilde{q}_E}(t)$ *and* $s \xrightarrow{\tau} s'$ *where*

$$s' = \oplus_{\substack{i\in I\\ k\in K}}\ g'_{ik}\ ([\widetilde{q}_P\widetilde{q}'_R\widetilde{q}_E \mapsto |\psi_{ik}\rangle]; \widetilde{q}_P, \widetilde{q}'_R; \lambda\widetilde{x}'.R'; \widetilde{v}_{R_k})$$

then there exists

$$t' = \oplus_{\substack{j\in J\\ k\in K}}\ h'_{jk}\ ([\widetilde{q}_Q\widetilde{q}'_R\widetilde{q}_E \mapsto |\phi_{jk}\rangle]; \widetilde{q}_Q, \widetilde{q}'_R; \lambda\widetilde{x}'.R'; \widetilde{v}_{R_k})$$

such that $t \xrightarrow{\tau} t'$ *and* $\rho^{\widetilde{q}'_R\widetilde{q}_E}(s') = \rho^{\widetilde{q}'_R\widetilde{q}_E}(t')$.

Proof. By induction on the derivation of $s \xrightarrow{\tau} s'$. Only the case L-Expr is non-trivial. □

The next two lemmas prove some simple results which will be used in the proof of Theorem 4.14.

Lemma 4.12. *Let* $s = \oplus_{i\in I}\ g_i\ s_i$ *and* $s' = \oplus_{i\in I}\ g_i\ s'_i$ *then* $s \xrightarrow{\alpha} s'$ *if and only if* $\forall i \in I.(s_i \xrightarrow{\alpha} s'_i)$ *for* $\alpha \in \{\cdot?[\cdot], \tau\}$.

Proof. By induction on the derivation of $s \xrightarrow{\alpha} s'$. This is because the process structure is constant for all $i \in I$. □

Lemma 4.13. *Let* $s_j = \oplus_{i\in I_j}\ g_{ij}\ (\sigma_{ij}; \omega; \lambda\widetilde{y}.P; \widetilde{v}_{ij})$ *and* $s_{ij} = (\sigma_{ij}; \omega; P\{\widetilde{v}_{ij}/\widetilde{y}\})$*. Then* $\forall j \in J, i \in I_j.(s_{ij} \xrightarrow{c?[\widetilde{u}_j,\widetilde{r}]}_p s'_{ij})$ *if and only if* $\forall j \in J.(s_j \xrightarrow{c?[\widetilde{u}_j,\widetilde{r}]}_p s'_j)$.

Proof. By induction on the derivation of $s_j \overset{c?[\widetilde{u}_j,\widetilde{r}]}{\longrightarrow}_p s'_j$. If the transition is derived from P-In then by L-In we have $\forall j \in J, i \in I_j.((\sigma_{ij};\omega;P) \overset{c?[\widetilde{u}_j,\widetilde{r}]}{\longrightarrow} (\sigma_{ij};\omega';P'))$ and by Lemma 4.12 we have $\forall j \in J$

$$\oplus_{i\in I_j}\ g_{ij}\ (\sigma_{ij};\omega;\lambda\widetilde{y}.P;\widetilde{v}_{ij}) \overset{c?[\widetilde{u}_j,\widetilde{r}]}{\longrightarrow} \oplus_{i\in I_j}\ g_{ij}\ (\sigma_{ij};\omega;\lambda\widetilde{y}.P';\widetilde{v}_{ij})\ .$$

The cases for P-Par, P-Sum and P-Res are similar, making use of L-Par, L-Sum and L-Res respectively. The argument is easily reversed to obtain the opposite direction. □

We are now in a position to prove that bisimilarity is preserved by parallel composition. To prove this, we will define an equivalence relation that contains the pair $((\sigma;\emptyset;P \parallel R),(\sigma;\emptyset;Q \parallel R))$, and that is closed under transitions from these configurations. After proving that bisimilarity of configurations is preserved by parallel composition, we can use the result to prove that bisimilarity of processes is preserved by parallel composition (Theorem 4.15).

Theorem 4.14 (Parallel preservation for configurations)**.** *Assume that* $\Gamma \vdash P$, $\Gamma \vdash Q$, $\Gamma \vdash P \parallel R$, *and* $\Gamma \vdash Q \parallel R$. *If* $(\sigma;\emptyset;P) \leftrightarrows (\sigma;\emptyset;Q)$ *then* $(\sigma;\emptyset;P \parallel R) \leftrightarrows (\sigma;\emptyset;Q \parallel R)$.

Proof. This proof is structured as follows. First, we introduce the notational conventions that will be used in this proof. We define an equivalence relation $\mathcal{R}$ on general configurations, in which the pair $((\sigma;\emptyset;P \parallel R),(\sigma;\emptyset;Q \parallel R))$ from the statement is a particular case. The remainder of the proof is dedicated to proving that $\mathcal{R}$ is a probabilistic branching bisimulation.

Let P, Q, R be general processes and assume that $\Gamma;\widetilde{q}_P \vdash P$, $\Gamma;\widetilde{q}_Q \vdash Q$, $\Gamma;\widetilde{q}_P,\widetilde{q}_R \vdash P \parallel R$, and $\Gamma;\widetilde{q}_Q,\widetilde{q}_R \vdash Q \parallel R$. Let K be an arbitrary indexing set. For each $k \in K$, let s_k and t_k be configurations given by

$$\begin{aligned} s_k &= \oplus_{i\in I_k}\ g_{ik}\ (\sigma_{ik};\widetilde{q}_P;\lambda\widetilde{x}_P.P;\widetilde{v}_{P_{ik}}) \\ t_k &= \oplus_{j\in J_k}\ h_{jk}\ (\tau_{jk};\widetilde{q}_Q;\lambda\widetilde{x}_Q.Q;\widetilde{v}_{Q_{jk}}) \end{aligned}$$

where $\sigma_{ik} = [\widetilde{q}_P\widetilde{q}_R\widetilde{q}_E \mapsto |\psi_{ik}\rangle]$ and $\tau_{jk} = [\widetilde{q}_Q\widetilde{q}_R\widetilde{q}_E \mapsto |\phi_{jk}\rangle]$, and where $\widetilde{q}_E$ are qubits in the environment and for each $k \in K$, $\rho_E(s_k) = \rho_E(t_k)$.

We use the convention that configurations su and tu are defined in relation to $\{s_k\}$ and $\{t_k\}$ where

$$\begin{aligned} su &= \oplus_{\substack{i\in I_k \\ k\in K}}\ f_k g_{ik}\ (\sigma_{ik};\widetilde{q}_P,\widetilde{q}_R;\lambda\widetilde{x}_P\widetilde{x}_R.P \parallel R;\widetilde{v}_{P_{ik}},\widetilde{v}_{R_k}) \\ tu &= \oplus_{\substack{j\in J_k \\ k\in K}}\ f_k h_{jk}\ (\tau_{jk};\widetilde{q}_Q,\widetilde{q}_R;\lambda\widetilde{x}_Q\widetilde{x}_R.Q \parallel R;\widetilde{v}_{Q_{jk}},\widetilde{v}_{R_k})\ . \end{aligned}$$

and $\{f_k\}$ is a set of weights. Following this convention, the configurations $\{s'_k\}$, $\{t'_k\}$, su' and tu', for example, are related in the same manner.

We use the convention that variables $\widetilde{x}_P$ (respectively $\widetilde{x}_Q$, $\widetilde{x}_R$) appear only in the process P (respectively Q, R). Therefore we are able to use the fact that configurations $(\sigma;\omega;\lambda\widetilde{x}_P\widetilde{x}_R.P;\widetilde{v}_P,\widetilde{v}_R)$ and $(\sigma;\omega;\lambda\widetilde{x}_P.P;\widetilde{v}_P)$ are structurally congruent; this is used implicitly throughout the proof.

Now define an equivalence relation $\mathcal{R}_1$ as

$$\mathcal{R}_1 = \{(su, tu) \mid \forall k \in K.(s_k \leftrightarrows t_k)\} \ .$$

Then define $\mathcal{R}$ to include probabilistic distributions, where

$$\mathcal{R} = \{(\boxplus_{m\in M}\ p_m \bullet s_m, \boxplus_{m\in M}\ p_m \bullet t_m) \mid \forall m \in M.(s_m, t_m) \in \mathcal{R}_1\} \ .$$

Now we prove that $\mathcal{R}$ is a probabilistic branching bisimulation.

By case analysis of the possible transitions of su; there are 9 cases to consider, namely an internal transition by P, output by P, input by P, communication from P, the respective transitions by R, and probabilistic transitions. In this proof we will use the convention that $s = \oplus_{k\in K}\ f_k\ s_k$ and $su = \oplus_{k\in K}\ f_k\ su_k$ in order to simplify the notation, and Theorem 4.4 (type preservation) will be used implicitly to ensure that the typing conditions remain satisfied.

Internal transition by P: If $su \xrightarrow{\tau} su'$ then by L-Par we have the hypothesis $s \xrightarrow{\tau} s'$ where

$$\begin{aligned} s' &= \oplus_{\substack{i\in I'_k\\ k\in K}}\ f_k g'_{ik}\ (\sigma'_{ik};\widetilde{q}'_P,\widetilde{q}_R;\lambda\widetilde{x}'_P.P';\widetilde{v}'_{P_{ik}}) \text{ and} \\ su' &= \oplus_{\substack{i\in I'_k\\ k\in K}}\ f_k g'_{ik}\ (\sigma'_{ik};\widetilde{q}'_P,\widetilde{q}_R;\lambda\widetilde{x}'_P\widetilde{x}_R.P' \parallel R;\widetilde{v}'_{P_{ik}},\widetilde{v}_{R_k}) \ . \end{aligned}$$

Lemma 4.12 gives $\forall k \in K.(s_k \xrightarrow{\tau} s'_k)$. Then, for each $k \in K$, because $s_k \leftrightarrows t_k$ there exist configurations t'_k, t''_k such that $t_k \Longrightarrow t'_k \xrightarrow{\tau}{}^{+} t''_k$ with $s_k \leftrightarrows t'_k$ and $s'_k \leftrightarrows t''_k$. Therefore by Lemma 4.12 we have $t \Longrightarrow t' \xrightarrow{\tau}{}^{+} t''$ where

$$\begin{aligned} t' &= \oplus_{\substack{j\in J'_k\\ k\in K}}\ f_k h'_{jk}\ (\tau'_{jk};\widetilde{q}'_Q,\widetilde{q}_R;\lambda\widetilde{x}'_Q.Q';\widetilde{v}'_{Q_{jk}}) \text{ and} \\ t'' &= \oplus_{\substack{j\in J''_k\\ k\in K}}\ f_k h''_{jk}\ (\tau''_{jk};\widetilde{q}''_Q,\widetilde{q}_R;\lambda\widetilde{x}''_Q.Q';\widetilde{v}'_{Q_{jk}}) \ . \end{aligned}$$

By L-Par we obtain the transitions $tu \Longrightarrow tu' \xrightarrow{\tau}{}^{+} tu''$ where

$$\begin{aligned} tu' &= \oplus_{\substack{j\in J'_k\\ k\in K}}\ f_k h'_{jk}\ (\tau'_{jk};\widetilde{q}'_Q,\widetilde{q}_R;\lambda\widetilde{x}'_Q\widetilde{x}_R.Q' \parallel R;\widetilde{v}'_{Q_{jk}},\widetilde{q}_{R_k}) \text{ and} \\ tu'' &= \oplus_{\substack{j\in J''_k\\ k\in K}}\ f_k h''_{jk}\ (\tau''_{jk};\widetilde{q}''_Q,\widetilde{q}_R;\lambda\widetilde{x}''_Q\widetilde{x}_R.Q' \parallel R;\widetilde{v}'_{Q_{jk}},\widetilde{q}_{R_k}) \ . \end{aligned}$$

Lemma 4.10 gives for each $k \in K$, $\rho_E(s_k) = \rho_E(s'_k)$ and $\rho_E(t_k) = \rho_E(t'_k) = \rho_E(t''_k)$ hence $\rho_E(s_k) = \rho_E(t'_k)$ and $\rho_E(s'_k) = \rho_E(t''_k)$. Therefore $(su, tu') \in \mathcal{R}$ and $(su', tu'') \in$

$\mathcal{R}$.

Internal transition by R: The transition $su \xrightarrow{\tau} su'$ has the hypothesis $u_1 \xrightarrow{\tau} u_1'$ where

$$u_1 = \oplus_{\substack{i\in I_k\\ k\in K}} f_k g_{ik}\ (\sigma_{ik}; \widetilde{q}_P, \widetilde{q}_R; \lambda\widetilde{x}_R.R; \widetilde{v}_{R_k}) \text{ and}$$
$$u_1' = \oplus_{\substack{i\in I_k\\ k\in K'}} f_k' g_{ik}\ (\sigma_{ik}'; \widetilde{q}_P, \widetilde{q}_R'; \lambda\widetilde{x}_R'.R'; \widetilde{v}_{R_k}')\ .$$

By Lemma 4.11 there exists u_2' such that $u_2 \xrightarrow{\tau} u_2'$ where

$$u_2 = \oplus_{\substack{j\in J_k\\ k\in K}} f_k h_{jk}\ (\tau_{jk}; \widetilde{q}_Q, \widetilde{q}_R; \lambda\widetilde{x}_R.R; \widetilde{v}_{R_k}),$$
$$u_2' = \oplus_{\substack{j\in J_k\\ k\in K'}} f_k' h_{jk}\ (\tau_{jk}'; \widetilde{q}_Q, \widetilde{q}_R'; \lambda\widetilde{x}_R'.R'; \widetilde{v}_{R_k}')$$

and $\rho^{\widetilde{q}_R'\widetilde{q}_E}(u_1') = \rho^{\widetilde{q}_R\widetilde{q}_E}(u_2')$. By L-Par we have $tu \xrightarrow{\tau} tu'$.

Let $s_k' = \oplus_{i\in I_k}\ g_{ik}\ (\sigma_{ik}'; \widetilde{q}_P; \lambda\widetilde{x}_P.P; \widetilde{v}_{ik})$ and $t_k' = \oplus_{j\in J_k}\ h_{jk}\ (\tau_{jk}'; \widetilde{q}_Q; \lambda\widetilde{x}_Q.Q; \widetilde{v}_{jk})$. We must show that $\forall k \in K.(s_k' \leftrightarrows t_k')$. It is only necessary to consider the possible cases for the derivation of $u_i \xrightarrow{\tau} u_i'$ in which the quantum state is altered; these are R-Trans, R-Measure and L-Qbit (in all other cases $s_k = s_k'$ and $t_k = t_k'$).

- R-Trans: For σ_{ik}', we have $[\widetilde{q}_P\widetilde{q}_R\widetilde{q}_E \mapsto |\psi_{ik}'\rangle = (I_P \otimes U \otimes I_E)|\psi_{ik}\rangle]$ for some unitary operator U and where I_P and I_E denote the identity operators on qubits $\widetilde{q}_P$ and $\widetilde{q}_E$ respectively. Similarly, for τ_{jk}' we have $[\widetilde{q}_Q\widetilde{q}_R\widetilde{q}_E \mapsto |\phi_{jk}'\rangle = (I_Q \otimes U \otimes I_E)|\phi_{jk}\rangle]$. Now define a relation $\mathcal{R}_u$ such that $(s_k', t_k') \in \mathcal{R}_u$ if $s_k \leftrightarrows t_k$ and $\rho(s_k') = (I_P \otimes U \otimes I_E)^\dagger \rho(s_k)(I_P \otimes U \otimes I_E)$ and $\rho(t_k') = (I_Q \otimes U \otimes I_E)^\dagger \rho(t_k)(I_Q \otimes U \otimes I_E)$.

  If $s_k' \xrightarrow{\tau} s_k''$ then, by a straightforward induction on the derivation, we have $s_k \xrightarrow{\tau} s_k'''$ and $\rho(s_k'') = (I_P' \otimes U \otimes I_E)^\dagger \rho(s_k''')(I_P' \otimes U \otimes I_E)$. Because $s_k \leftrightarrows t_k$ we have $t_k \Longrightarrow t_k'' \xrightarrow{\tau}{}^{+} t_k'''$ and $s_k \leftrightarrows t_k''$ and $s_k'' \leftrightarrows t_k'''$. By induction on the derivation of each transition in this sequence, we obtain $t_k' \Longrightarrow t_k'''' \xrightarrow{\tau}{}^{+} t_k'''''$ where $\rho(t_k'''') = (I_Q'' \otimes U \otimes I_E)^\dagger \rho(t_k'')(I_Q'' \otimes U \otimes I_E)$ and $\rho(t_k''''') = (I_Q''' \otimes U \otimes I_E)^\dagger \rho(t_k''')(I_Q''' \otimes U \otimes I_E)$. Therefore $(s_k', t_k'''') \in \mathcal{R}_u$ and $(s_k'', t_k''''') \in \mathcal{R}_u$.

  If $s_k' \xrightarrow{c?[\widetilde{q},\widetilde{v}]} s_k''$ then similar reasoning applies as in the previous case.

  If $s_k' \xrightarrow{c![V,\widetilde{r}]} \boxplus_m\ p_m \bullet s_{km}''$ then $s_k \xrightarrow{c![V,\widetilde{r}]} \boxplus_m\ p_m \bullet s_{km}'''$ and $\rho(s_{km}'') = (I_P' \otimes U \otimes I_E)^\dagger \rho(s_{km}''')(I_P' \otimes U \otimes I_E)$ and then we have $\rho^{\widetilde{r}\widetilde{q}_R\widetilde{q}_E}(s_{km}'') = (I_{\widetilde{r}} \otimes U \otimes I_E)^\dagger \rho^{\widetilde{r}\widetilde{q}_R\widetilde{q}_E}(s_{km}''')(I_{\widetilde{r}} \otimes U \otimes I_E)$. Because $s_k \leftrightarrows t_k$ we have $t_k \Longrightarrow t_k'' \xrightarrow{c![V,\widetilde{s}]} \boxplus_m\ p_m \bullet t_{km}'''$ and $s_k \leftrightarrows t_k''$ and $\forall m.(s_{km}'' \leftrightarrows t_{km}''')$. By induction on the derivation of each transition in this sequence, we obtain $t_k' \Longrightarrow t_k'''' \xrightarrow{c![V,\widetilde{s}]} \boxplus_m\ p_m \bullet t_{km}'''''$ where $\rho(t_k'''') = (I_Q'' \otimes U \otimes I_E)^\dagger \rho(t_k'')(I_Q'' \otimes U \otimes I_E)$ and $\forall m.(\rho(t_{km}''''') = (I_Q''' \otimes U \otimes I_E)^\dagger \rho(t_{km}''')(I_Q''' \otimes U \otimes I_E))$. Therefore $\rho^{\widetilde{s}\widetilde{q}_R\widetilde{q}_E}(t_{km}''''') = (I_{\widetilde{s}} \otimes U \otimes I_E)^\dagger \rho^{\widetilde{s}\widetilde{q}_R\widetilde{q}_E}(t_{km}''')(I_{\widetilde{s}} \otimes U \otimes I_E)$ and because $\rho^{\widetilde{r}\widetilde{q}_R\widetilde{q}_E}(s_{km}''') = \rho^{\widetilde{s}\widetilde{q}_R\widetilde{q}_E}(t_{km}''')$ we have

$\rho^{\widetilde{r}\widetilde{q}_R\widetilde{q}_E}(s''_{km}) = \rho^{\widetilde{s}\widetilde{q}_R\widetilde{q}_E}(t''''_{km})$. Therefore $(s'_k, t''''_k) \in \mathcal{R}_u$ and $\forall m.(s''_{km}, t''''_{km}) \in \mathcal{R}_u$. We find that $\mathcal{R}_u$ is a probabilistic branching bisimulation, hence $s'_k \leftrightarroweq t'_k$.

- R-MEASURE: We have a set of measurement operators $\{M_m\}$ such that $\rho(s'_k) = \sum_m f_m(I_P \otimes M_m \otimes I_E)^\dagger \rho(s_k)(I_P \otimes M_m \otimes I_E)$ and $\rho(t'_k) = \sum_m f_m(I_Q \otimes M_m \otimes I_E)^\dagger \rho(t_k)(I_Q \otimes M_m \otimes I_E)$. We construct a relation $\mathcal{R}_m$ such that $(s'_k, t'_k) \in \mathcal{R}_m$ if $s_k \leftrightarroweq t_k$ and $\rho(s'_k) = \sum_m f_m(I_P \otimes M_m \otimes I_E)^\dagger \rho(s_k)(I_P \otimes M_m \otimes I_E)$ and $\rho(t'_k) = \sum_m f_m(I_Q \otimes M_m \otimes I_E)^\dagger \rho(t_k)(I_Q \otimes M_m \otimes I_E)$. By similar reasoning to the previous case, we find that $\mathcal{R}_m$ is a bisimulation, hence $s_k \leftrightarroweq t_k$.

- L-QBIT: We have the relationships $\rho(s'_k) = \rho(s_k) \otimes |0\rangle\langle 0|$ and $\rho(t'_k) = \rho(t_k) \otimes |0\rangle\langle 0|$. We construct a relation and follow similar reasoning to the previous cases.

Communication from P: The derivation by L-COM is

$$\frac{\forall k \in K, i \in I_k.((s_{ik} \xrightarrow{c![\widetilde{u}_{ik},\widetilde{q}]}_p s'_{ik}) \quad (u_{ik} \xrightarrow{c?[\widetilde{u}_{ik},\widetilde{q}]}_p u'_{ik}))}{su \xrightarrow{\tau} su'}$$

where

$$\begin{aligned} s_{ik} &= (\sigma_{ik}; \widetilde{q}_P, \widetilde{q}_R; P\{\widetilde{v}_{P_{ik}}/\widetilde{x}_P\}), \\ s'_{ik} &= (\sigma_{ik}; \widetilde{q}'_P, \widetilde{q}_R; P'\{\widetilde{v}_{P_{ik}}/\widetilde{x}_P\}), \\ u_{ik} &= (\sigma_{ik}; \widetilde{q}_P, \widetilde{q}_R; R\{\widetilde{v}_{R_k}/\widetilde{x}_R\}), \\ u'_{ik} &= (\sigma_{ik}; \widetilde{q}_P, \widetilde{q}'_R; R'\{\widetilde{v}_{R_k}/\widetilde{x}_R\}) \end{aligned}$$

and

$$su' = \oplus_{\substack{i \in I_k \\ k \in K}} f_k g_{ik} \ (\sigma_{ik}; \widetilde{q}'_P, \widetilde{q}'_R; \lambda \widetilde{x}_P \widetilde{x}_R.P' \parallel R'; \widetilde{v}_{P_{ik}}, \widetilde{v}_{R_k}) \ .$$

For each $k \in K$, we derive by L-OUT the transition $(s_k \xrightarrow{c![U_k,\widetilde{q}]} s'_{k_o})$ where $U_k = \{u_{ik} \mid i \in I_k\}$ and

$$s'_{k_o} = \boxplus_{m \in M_k} p_m \bullet s_{km_o} \text{ and } s_{km_o} = (\oplus_{i \in I_{km}} \frac{g_{ik}}{p_m} \ (\sigma'_{ik}; \widetilde{q}'_P; \lambda \widetilde{x}_P.P'; \widetilde{v}_{P_{ik}}) \ .$$

For each $k \in K$, because $s_k \leftrightarroweq t_k$ we get t'_k, t''_{k_o} such that $t_k \Longrightarrow t'_k \xrightarrow{c![U_k,\widetilde{r}]} t''_{k_o}$ where

$$\begin{aligned} t'_k &= \oplus_{j \in J'_k} h'_{jk} \ (\tau'_{jk}; \widetilde{q}'_Q; \lambda \widetilde{x}'_Q.Q'; \widetilde{v}'_{Q_{jk}}), \\ t''_{k_o} &= \boxplus_{m \in M_k} p_m \bullet t''_{km_o}, \\ t''_{km_o} &= \oplus_{j \in J'_{km}} h'_{jk} \ (\tau''_{jk}; \widetilde{q}''_Q; \lambda \widetilde{x}'_Q.Q''; \widetilde{v}'_{Q_{jk}}) \end{aligned}$$

and $s_k \leftrightarroweq t'_k$ and for each $m \in M_k$, $s'_{km_o} \leftrightarroweq t''_{km_o}$ and $\rho_E(s_{km_o}) = \rho_E(t'_{km_o})$. Applying Lemma 4.12 to each step in $t_k \Longrightarrow t'_k$ gives $tu \Longrightarrow tu'$. By L-COM we can derive the transition $tu' \xrightarrow{\tau} tu''$.

Now, by Lemma 4.10 we have for each $k \in K$, $\rho_E(t_k) = \rho_E(t'_k)$, therefore it follows that $\rho_E(s_k) = \rho_E(t'_k)$ and because $s_k \leftrightarrows t'_k$ we have $(su, tu') \in \mathcal{R}$. By convention we have

$$s'_k = \oplus_{i \in I_k}\ g_{ik}\ (\sigma_{ik}; \widetilde{q}'_P; \lambda \widetilde{x}_P.P'; \widetilde{v}_{P_{ik}}) \text{ and}$$
$$t''_k = \oplus_{j \in J'_k}\ h'_{jk}\ (\tau'_{jk}; \widetilde{q}''_Q; \lambda \widetilde{x}_Q.Q''; \widetilde{v}'_{Q_{jk}})$$

where σ_{ik} and σ'_{ik} (respectively τ'_{jk} and τ''_{jk}) differ by the permutation and renaming applied by L-Out. Because for each $m \in M_k$, $s'_{km_o} \leftrightarrows t''_{km_o}$, we have

$$\oplus_{i \in I_k}\ g_{ik}\ (\sigma'_{ik}; \widetilde{q}'_P; \lambda \widetilde{x}_P.P'; \widetilde{v}_{P_{ik}}) \leftrightarrows \oplus_{j \in J'_k}\ h'_{jk}\ (\tau''_{jk}; \widetilde{q}''_Q; \lambda \widetilde{x}'_Q.Q''; \widetilde{x}'_{Q_{jk}})$$

therefore it follows that $s'_k \leftrightarrows t''_k$. It follows from $\rho_E(s_k) = \rho_E(t'_k)$ that $\rho_E(s'_k) = \rho_E(t''_k)$, therefore $(su', tu'') \in \mathcal{R}$.

Communication from R: This transition is derived by L-Com:

$$\frac{\forall k \in K, i \in I_k.(s_{ik} \xrightarrow{c?[\widetilde{u}_k, \widetilde{r}]}_p s'_{ik} \quad u_{ik} \xrightarrow{c![\widetilde{u}_k], \widetilde{r}}_p u'_{ik})}{su \xrightarrow{\tau} su'}\ .$$

Because the output is from R, the classical values $\widetilde{u}_k$ that are transferred in the communication must be dependent on the index k and be independent of i. We rewrite the configurations so that $\widetilde{u}_k$ are copies of the respective values in $\widetilde{v}_{R_k}$; this enables us to maintain the distinction between variables appearing in the respective processes P, Q and R after the communication. Therefore we have

$$su = \oplus_{\substack{i \in I_k \\ k \in K}}\ f_k g_{ik}\ (\sigma_{ik}; \widetilde{q}_P, \widetilde{q}_R; \lambda \widetilde{x}_P \widetilde{x}_R \widetilde{y}.P \parallel R; \widetilde{v}_{P_{ik}}, \widetilde{v}_{R_k}, \widetilde{u}_k)$$

and

$$su' = \oplus_{\substack{i \in I_k \\ k \in K}}\ f_k g_{ik}\ (\sigma_{ik}; \widetilde{q}'_P, \widetilde{q}'_R; \lambda \widetilde{x}_P \widetilde{x}_R \widetilde{y}.P' \parallel R'; \widetilde{v}_{P_{ik}}, \widetilde{v}_{R_k}, \widetilde{u}_k)\ .$$

For each $k \in K$, because $\forall i \in I_k.(s_{ik} \xrightarrow{c?[\widetilde{u}_k, \widetilde{r}]}_p s'_{ik})$ we obtain by Lemma 4.13 that $s_k \xrightarrow{c?[\widetilde{u}_k, \widetilde{r}]} s'_k$. Furthermore, because $s_k \leftrightarrows t_k$, there exist t'_k and t''_k such that $t_k \Longrightarrow t'_k \xrightarrow{c?[\widetilde{u}_k, \widetilde{r}]} t''_k$ where $s_k \leftrightarrows t'_k$ and $s'_k \leftrightarrows t''_k$. Then by applying L-Par to each step of the transition $t_k \Longrightarrow t'_k$ we obtain $tu \Longrightarrow tu'$, and applying Lemma 4.13 to the transition $t'_k \xrightarrow{c?[\widetilde{u}_k, \widetilde{r}]} t''_k$ gives $\forall j \in J'_k.(t'_{ik} \xrightarrow{c?[\widetilde{u}_k, \widetilde{r}]}_p t''_{ik})$. Therefore by L-Com we can derive the transition

$$\frac{\forall k \in K, j \in J'_k.(t'_{jk} \xrightarrow{c?[\widetilde{u}_k, \widetilde{r}]}_p t''_{jk} \quad u_{ik} \xrightarrow{c![\widetilde{u}_k, \widetilde{r}]}_p u'_{ik})}{tu' \xrightarrow{\tau} tu''}\ .$$

Using Lemma 4.10 we have $\rho_E(t_k) = \rho_E(t'_k)$, hence $\rho_E(s_k) = \rho_E(t'_k)$. Then we have $\rho_E(s'_k) = \mathrm{tr}_{\widetilde{r}} \rho_E(s_k)$ and $\rho_E(t''_k) = \mathrm{tr}_{\widetilde{r}} \rho_E(t'_k)$, from which we obtain $\rho_E(s'_k) = \rho_E(t''_k)$.

Therefore we have $(su, tu') \in \mathcal{R}$ and $(su', tu'') \in \mathcal{R}$ as required.

Output by P: If $su \xrightarrow{c![U,\widetilde{q}]} su'$ where

$$su' = \uplus_{m\in M}\ p_m \bullet \oplus_{\substack{i\in I_{km}\\ k\in K}} \frac{f_k g_{ik}}{p_m}\ (\sigma'_{ik}; \widetilde{q}'_P, \widetilde{q}_R; \lambda\widetilde{x}_P\widetilde{x}_R.P' \parallel R; \widetilde{v}_{P_{ik}}, \widetilde{v}_{R_k})$$

then the derivation by L-Out and P-Par has the hypothesis $\forall k \in K, i \in I_k.(s_{ik} \xrightarrow{c![\widetilde{u}_{ik},\widetilde{q}]}_p s'_{ik})$ where $U = \{\widetilde{u}_{ik} \mid i \in I_k, k \in K\}$ and

$$\begin{aligned} s_{ik} &= (\sigma_{ik}; \widetilde{q}_P, \widetilde{q}_R; P\{\widetilde{v}_{P_{ik}}/\widetilde{x}_P\}) \text{ and}\\ s'_{ik} &= (\sigma_{ik}; \widetilde{q}'_P, \widetilde{q}_R; P'\{\widetilde{v}_{P_{ik}}/\widetilde{x}_P\})\ . \end{aligned}$$

Then, for each $k \in K$, by L-Out we have $s_k \xrightarrow{c![U_k,\widetilde{q}]} s'_k$ where

$$\begin{aligned} s_k &= \oplus_{i\in I_k}\ g_{ik}\ s_{ik},\\ s'_k &= \uplus_{m\in M_k}\ p_{km} \bullet s'_{km} \text{ and}\\ s'_{km} &= \oplus_{i\in I_{km}}\ \frac{g_{ik}}{p_{km}}\ s'_{ik}\ . \end{aligned}$$

For each $k \in K$, because $s_k \leftrightarrows t_k$ then there exists t'_k, t''_k such that $t_k \Longrightarrow t'_k \xrightarrow{c![U_k,\widetilde{r}]} t''_k$ and $s_k \leftrightarrows t'_k$ and $t''_k = \uplus_{m\in M_k}\ p_{km} \bullet t''_{km}$ and $\forall m \in M_k.(s'_{km} \leftrightarrows t''_{km})$ and $\rho_E(s'_{km}) = \rho_E(t''_{km})$. Then, for each $k \in K$, the derivation of $t'_k \xrightarrow{c![U_k,\widetilde{r}]} t''_k$ gives the hypothesis $\forall j \in J_k.(t'_{jk} \xrightarrow{c![\widetilde{v}_{jk},\widetilde{r}]}_p t''_{jk})$ where

$$\begin{aligned} t'_k &= \oplus_{j\in J'_k}\ h'_{jk}\ (\tau'_{jk}; \widetilde{q}'_Q, \widetilde{q}_R; \lambda\widetilde{x}'_Q.Q'; \widetilde{v}'_{Q_{jk}}),\\ t''_{km} &= \oplus_{j\in J_{km}}\ \frac{h'_{jk}}{p_{km}}\ (\tau'_{jk}; \widetilde{q}''_Q, \widetilde{q}_R; \lambda\widetilde{x}'_Q.Q''; \widetilde{v}'_{Q_{jk}}),\\ t'_{jk} &= (\tau'_{jk}; \widetilde{q}'_Q, \widetilde{q}_R; Q'\{\widetilde{x}'_Q/\widetilde{v}'_{Q_{jk}}\}) \text{ and}\\ t''_{jk} &= (\tau'_{jk}; \widetilde{q}''_Q, \widetilde{q}_R; Q''\{\widetilde{x}'_Q/\widetilde{v}'_{Q_{jk}}\})\ . \end{aligned}$$

Now, applying Lemma 4.12 to each step in the transitions $t_k \Longrightarrow t'_k$ gives $tu \Longrightarrow tu'$ where

$$tu' = \oplus_{\substack{j\in J'_k\\ k\in K}}\ f_k h'_{jk}\ (\tau'_{jk}; \widetilde{q}'_Q, \widetilde{q}_R; \lambda\widetilde{x}'_Q\widetilde{x}_R.Q' \parallel R; \widetilde{v}'_{Q_{jk}}, \widetilde{v}_{R_k})\ .$$

Using P-Par and L-Out we can derive the transition $tu' \xrightarrow{c![U,\widetilde{r}]} tu''$ where

$$tu'' = \uplus_{m\in M}\ p_m \bullet \oplus_{\substack{j\in J'_{km}\\ k\in K}} \frac{f_k h'_{jk}}{p_m}\ (\tau''_{jk}; \widetilde{q}''_Q, \widetilde{q}_R; \lambda\widetilde{x}'_Q\widetilde{x}_R.Q'' \parallel R; \widetilde{v}'_{Q_{jk}}, \widetilde{v}_{R_k}),$$

noting that $p_m = \sum_{k\in K} p_{km}$. Let

$$su'_m = \oplus_{\substack{i\in I_{km}\\ k\in K}} \frac{f_k g_{ik}}{p_m} (\sigma'_{ik}; \widetilde{q}'_P, \widetilde{q}_R; \lambda\widetilde{x}_P\widetilde{x}_R.P' \parallel R; \widetilde{v}_{P_{ik}}, \widetilde{v}_{R_k}) \text{ and}$$
$$tu''_m = \oplus_{\substack{j\in J'_{km}\\ k\in K}} \frac{f_k h'_{jk}}{p_m} (\tau''_{jk}; \widetilde{q}''_Q, \widetilde{q}_R; \lambda\widetilde{x}'_Q\widetilde{x}_R.Q'' \parallel R; \widetilde{v}'_{Q_{jk}}, \widetilde{v}_{R_k})$$

then, for each $m \in M$ because $\forall k \in K.(s'_{km} \leftrightarrows t''_{km})$ and $\rho_E(s'_{km}) = \rho_E(t''_{km})$ we have $(su'_m, tu''_m) \in \mathcal{R}$. For each $k \in K$, using Lemma 4.10 we have $\rho_E(t_k) = \rho_E(t'_k)$, hence $\rho_E(s_k) = \rho_E(t'_k)$ and therefore $(su, tu') \in \mathcal{R}$.

Output by R: If $su \xrightarrow{c![U,\widetilde{r}]} su'$ then the derivation by L-Out and P-Par gives the hypothesis

$$\forall k \in K, i \in I_k,\ (\sigma_{ik}; \widetilde{q}_P, \widetilde{q}_R; R\{\widetilde{v}_{R_k}/\widetilde{x}_R\}) \xrightarrow{c![\widetilde{u}_k,\widetilde{r}]}_p (\sigma_{ik}; \widetilde{q}_P, \widetilde{q}'_R; R'\{\widetilde{v}_{R_l}/\widetilde{x}_R\})$$

where $U = \{\widetilde{u}_k\} = \{\widetilde{v}_m\}$ and each list of values $\widetilde{u}_k$ is only dependent on k since it must be contained within $\widetilde{v}_{R_k}$. Because these transitions are independent of the quantum state, we get

$$\forall k \in K, j \in J_k,\ (\tau_{jk}; \widetilde{q}_Q, \widetilde{q}_R; R\{\widetilde{v}_{R_k}/\widetilde{x}_R\}) \xrightarrow{c![\widetilde{u}_k,\widetilde{r}]}_p (\tau_{jk}; \widetilde{q}_Q, \widetilde{q}'_R; R'\{\widetilde{v}_{R_l}/\widetilde{x}_R\})\ .$$

By applying P-Par and L-Out we can derive the transition $tu \xrightarrow{c![U,\widetilde{r}]} tu'$ where

$$tu' = \boxplus_{m\in M}\ p'_m \bullet \oplus_{\substack{j\in J_k\\ k\in K_m}} \frac{f_k}{p'_m} h_{jk}\ (\tau'_{jk}; \widetilde{q}_Q, \widetilde{q}_R; \lambda\widetilde{x}_Q\widetilde{x}_R.Q \parallel R; \widetilde{v}_{Q_{jk}}, \widetilde{v}_{R_k})\ .$$

For each $m \in M$ let $K_m = \{k \mid \widetilde{u}_k = \widetilde{v}_m\}$, then we have

$$p_m = \sum_{k\in K_m} f_k \sum_{i\in I_k} g_{ik} = \sum_{k\in K_m} f_k = \sum_{k\in K_m} f_k \sum_{j\in J_k} h_{jk} = p'_m\ .$$

Let $su' = \boxplus_{m\in M}\ p_m \bullet su'_m$ and $tu' = \boxplus_{m\in M}\ p_m \bullet tu'_m$ and let Π be the permutation operator corresponding to the permutation $\widetilde{q}_R\widetilde{q}_E \mapsto \widetilde{q}'_R\widetilde{q}_E\widetilde{r}$ (this permutation is applied in the transformation from σ_{ik} to σ'_{ik} and from τ_{jk} to τ'_{jk} due to L-Out). Then we have $\rho^{\widetilde{q}'_R\widetilde{q}_E\widetilde{r}}(su'_m) = \sum_{k\in K_m} \frac{f_k}{p_m}(I_P \otimes \Pi)^\dagger \rho_E(s_k)(I_P \otimes \Pi)$ and $\rho^{\widetilde{q}'_R\widetilde{q}_E\widetilde{r}}(tu'_m) = \sum_{k\in K_m} \frac{f_k}{p_m}(I_Q \otimes \Pi)^\dagger \rho_E(t_k)(I_Q \otimes \Pi)$. Because for each $k \in K$, $\rho_E(s_k) = \rho_E(t_k)$ and $\rho_E(su'_m) = \mathrm{tr}_{\widetilde{q}'_R}(\rho^{\widetilde{q}'_R\widetilde{q}_E\widetilde{r}}(su'_m))$ and $\rho_E(tu'_m) = \mathrm{tr}_{\widetilde{q}'_R}(\rho^{\widetilde{q}'_R\widetilde{q}_E\widetilde{r}}(tu'_m))$ we have $\rho_E(su'_m) = \rho_E(tu'_m)$. Then, because for each $k \in K, (s_k \leftrightarrows t_k)$ we have $\forall m \in M.((su'_m, tu'_m) \in \mathcal{R})$.

Input by P: We have the transition $su \xrightarrow{c?[\widetilde{u},\widetilde{r}]} su'$ where

$$su' = \oplus_{\substack{i\in I_k\\ k\in K}}\ f_k g_{ik}\ (\sigma_{ik}; \widetilde{q}_P, \widetilde{r}, \widetilde{q}_R; \lambda\widetilde{x}_P\widetilde{x}_R.P' \parallel R; \widetilde{v}_{P_{ik}}, \widetilde{v}_{R_k})\ .$$

The derivation of this transition by L-Par gives the hypothesis $s \xrightarrow{c?[\widetilde{u},\widetilde{r}]} s'$ where

$$s = \oplus_{\substack{i\in I_k\\k\in K}} f_k g_{ik}\ (\sigma_{ik}; \widetilde{q}_P; \lambda\widetilde{x}_P.P; \widetilde{v}_{P_{ik}}) \text{ and}$$
$$s' = \oplus_{\substack{i\in I_k\\k\in K}} f_k g_{ik}\ (\sigma_{ik}; \widetilde{q}_P, \widetilde{r}; \lambda\widetilde{x}_P.P'; \widetilde{v}_{P_{ik}})\ .$$

Applying Lemma 4.12 gives $\forall k \in K.(s_k \xrightarrow{c?[\widetilde{u},\widetilde{r}]} s'_k)$ where

$$s'_k = \oplus_{i\in I_k}\ g_{ik}\ (\sigma_{ik}; \widetilde{q}_P, \widetilde{r}; \lambda\widetilde{x}_P.P'; \widetilde{v}_{P_{ik}})\ .$$

For each $k \in K$, because $s_k \leftrightarroweq t_k$ there exist configurations t'_k, t''_k such that $t_k \Longrightarrow t'_k \xrightarrow{c?[\widetilde{u},\widetilde{r}]} t''_k$ where $s_k \leftrightarroweq t'_k$ and $s'_k \leftrightarroweq t''_k$. We now apply Lemma 4.12 to these transitions to get $t \Longrightarrow t' \xrightarrow{c?[\widetilde{u},\widetilde{r}]} t''$. Applying L-Par then gives the required transitions $tu \Longrightarrow tu' \xrightarrow{c?[\widetilde{u},\widetilde{r}]} tu''$. For each $k \in K$ we have $\rho_E(s'_k) = \mathrm{tr}_{\widetilde{r}}(\rho_E(s_k))$ and $\rho_E(t''_k) = \mathrm{tr}_{\widetilde{r}}(\rho_E(t'_k))$ and (by Lemma 4.10) $\rho_E(t''_k) = \rho_E(t'_k)$, then because $s_k \leftrightarroweq t'_k$ and $s'_k \leftrightarroweq t''_k$ we have $(su, tu') \in \mathcal{R}$ and $(su', tu'') \in \mathcal{R}$.

Input by R: We have the transition $su \xrightarrow{c?[\widetilde{u},\widetilde{r}]} su'$ where

$$su' = \oplus_{\substack{i\in I_k\\k\in K}} f_k g_{ik}\ (\sigma_{ik}; \widetilde{q}_P, \widetilde{q}'_R, \lambda\widetilde{x}_P\widetilde{x}_R.P \parallel R'; \widetilde{v}_{P_{ik}}, \widetilde{v}_{R_k})$$

Then the derivation by L-Par gives the transition $u_1 \xrightarrow{c?[\widetilde{u},\widetilde{r}]} u'_1$ corresponding to the action of R in isolation, where

$$u_1 = \oplus_{\substack{i\in I_k\\k\in K}} f_k g_{ik}\ (\sigma_{ik}; \widetilde{q}_P, \widetilde{q}_R, \lambda\widetilde{x}_R.R; \widetilde{v}_{R_k}) \text{ and}$$
$$u'_1 = \oplus_{\substack{i\in I_k\\k\in K}} f_k g_{ik}\ (\sigma_{ik}; \widetilde{q}_P, \widetilde{q}'_R, \lambda\widetilde{x}_R.R'; \widetilde{v}_{R_k})\ .$$

Since this transition is independent from the quantum state we obtain the transition $u_2 \xrightarrow{c?[\widetilde{u},\widetilde{r}]} u'_2$ where

$$u_2 = \oplus_{\substack{j\in J_k\\k\in K}} f_k h_{jk}\ (\tau_{ik}; \widetilde{q}_Q, \widetilde{q}_R, \lambda\widetilde{x}_R.R; \widetilde{v}_{R_k}) \text{ and}$$
$$u'_2 = \oplus_{\substack{j\in J_k\\k\in K}} f_k h_{jk}\ (\tau_{ik}; \widetilde{q}_Q, \widetilde{q}'_R, \lambda\widetilde{x}_R.R'; \widetilde{v}_{R_k})\ .$$

Applying L-Par to this transition gives $tu \xrightarrow{c?[\widetilde{u},\widetilde{r}]} tu'$ where

$$tu' = \oplus_{\substack{j\in J_k\\k\in K}} f_k h_{jk}\ (\tau_{ik}; \widetilde{q}_Q, \widetilde{q}_R, \lambda\widetilde{x}_Q\widetilde{x}'_R.Q \parallel R'; \widetilde{v}_{Q_{jk}}, \widetilde{v}_{R_k})\ .$$

Because the qubits $\widetilde{r}$ are contained within $\widetilde{q}_E$, we have $\widetilde{q}'_R = \widetilde{q}_R, \widetilde{r}$ and $\widetilde{q}_E = \widetilde{q}'_E, \widetilde{r}$. Therefore $\rho^{\widetilde{q}_R\widetilde{q}_E}(su) = \rho^{\widetilde{q}'_R\widetilde{q}'_E}(su')$ and $\rho^{\widetilde{q}_R\widetilde{q}_E}(tu) = \rho^{\widetilde{q}'_R\widetilde{q}'_E}(tu')$. Because $\forall k \in K.(s_k \leftrightarroweq t_k)$, we have $(su', tu') \in \mathcal{R}$.

Probabilistic configurations: If $su = \boxplus_{m \in M}\ p_m \bullet su_m$ then by the definition of $\mathcal{R}$ we must have $tu = \boxplus_{m \in M}\ p_m \bullet tu_m$ where for each $m \in M$, $(su_m, tu_m) \in \mathcal{R}$. Therefore, for each $m \in M$ we have $\mu(su, su_m) = p_m = \mu(tu, tu_m)$. The respective sets $\{su_m\}$ and $\{tu_m\}$ are exhaustive since $\sum_{m \in M} p_m = 1$, hence we have $\forall D \in \mathcal{S}/\mathcal{R}.(\mu(su, D) = \mu(tu, D))$. □

Theorem 4.15 (Parallel Preservation)**.** *If $P \leftrightarrows Q$ then for any process R such that $\Gamma \vdash P \parallel R$ and $\Gamma \vdash Q \parallel R$ then $P \parallel R \leftrightarrows Q \parallel R$.*

Proof. Because $P \leftrightarrows Q$ we have for all σ, $(\sigma; \emptyset; P) \leftrightarrows (\sigma; \emptyset; Q)$. Define a relation $\mathcal{R}$ according to the statement of Theorem 4.14; using the same notation, if $s_k = (\sigma; \emptyset; P)$ and $t_k = (\sigma; \emptyset; Q)$ then $su = (\sigma; \emptyset; P \parallel R)$ and $tu = (\sigma; \emptyset; Q \parallel R)$. Then we have for all σ, $((\sigma; \emptyset; P \parallel R), (\sigma; \emptyset; Q \parallel R)) \in \mathcal{R}$. By Theorem 4.14 $\mathcal{R}$ is a bisimulation, hence $P \parallel R \leftrightarrows Q \parallel R$. □

We now consider preservation with respect to other process constructions.

Lemma 4.16. *Probabilistic branching bisimilarity is preserved by output prefix, action prefix, channel restriction and non-deterministic choice.*

Proof. This proof consists of a subset of the cases from the proof of Lemma 4.19. □

We have now shown that probabilistic branching bisimilarity is preserved by all process constructs except input and qubit declaration. Following the approach used for the π-calculus in [Sangiorgi and Walker 2001] we first consider a *non-input congruence*, however for our language we must also exclude qubit declaration, hence we consider *non-input, non-qubit congruence.*

Theorem 4.17 (Probabilistic branching bisimilarity is a non-input, non-qubit congruence)**.** *If $P \leftrightarrows Q$ and for any non-input, non-qubit context C if $\Gamma \vdash C[P]$ and $\Gamma \vdash C[Q]$ then $C[P] \leftrightarrows C[Q]$.*

Proof. Follows directly from Theorem 4.15 and Lemma 4.16. □

4.3.2 Full Probabilistic Branching Bisimilarity

Probabilistic branching bisimilarity is not a congruence because it is not preserved by substitution. We can therefore define a stronger relation, *full probabilistic branching bisimilarity*, in which equivalence must also be preserved by substitutions.

Definition 4.10 (Full probabilistic branching bisimilarity)**.** Processes P and Q are *full probabilistic branching bisimilar*, denoted $P \leftrightarrows^c Q$, if for any substitution $\kappa = \{\widetilde{u}, \widetilde{q}/\widetilde{x}\}$ and for any quantum state σ, $(\sigma; \widetilde{q}; P\kappa) \leftrightarrows (\sigma; \widetilde{q}; Q\kappa)$.

We now prove that full probabilistic branching bisimilarity is preserved by all process constructs. We proved that probabilistic branching bisimilarity is preserved by parallel composition in Theorem 4.14, and note that this is independent of substitutions. Lemma 4.19 considers the other process constructs.

First we prove the following lemma, which will be used in the proof of Lemma 4.19.

Lemma 4.18. *If* $\forall i \in I.((\sigma_i;\omega;P) \leftrightarroweq (\sigma_i;\omega;Q))$ *and* $\sum_{i\in I} g_i = 1$ *then* $\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;P) \leftrightarroweq \oplus_{i\in I}\ g_i\ (\sigma_i;\omega;Q)$.

Proof. There is a bisimulation $\mathcal{R}_1$ such that $\forall i \in I, ((\sigma_i;\omega;P),(\sigma_i;\omega;Q)) \in \mathcal{R}_1$. Now define a relation $\mathcal{R}_2$ as

$$\mathcal{R}_2 = \{(\oplus_{\substack{i\in I\\ j\in J_i}}\ f_i g_{ij}\ (\sigma_{ij};\omega_P;\lambda\widetilde{y}_P.P;\widetilde{v}_{P_{ij}}), \oplus_{\substack{i\in I\\ k\in K_i}}\ f_i h_{ik}\ (\tau_{ik};\omega_Q;\lambda\widetilde{y}_Q.Q;\widetilde{v}_{Q_{ik}})) \mid \forall i \in I.(((\sigma_{ij};\omega_P;\lambda\widetilde{y}_P.P;\widetilde{v}_{P_{ij}}),(\tau_{ik};\omega_Q;\lambda\widetilde{y}_Q.Q;\widetilde{v}_{Q_{ik}})) \in \mathcal{R}_1)\}\ .$$

Then extend this relation to include probabilistic configurations:

$$\mathcal{R}_3 = \{(\boxplus_{m\in M}\ p_m \bullet s_m, \boxplus_{m\in M}\ p_m \bullet t_m) \mid \forall m \in M.((s_m,t_m) \in \mathcal{R}_2)\}\ .$$

We now show that $\mathcal{R}_2 \cup \mathcal{R}_3$ is a bisimulation.

For $(s,t) \in \mathcal{R}_2$, if $s \xrightarrow{\alpha} s'$ where

$$s' = \oplus_{\substack{i\in I\\ j\in J'_i}}\ f_i g'_{ij}\ (\sigma'_{ij};\omega'_P;\lambda\widetilde{y}'_P.P';\widetilde{v}'_{P_{ij}})$$

then by Lemma 4.12 we have $\forall i \in I.(s_i \xrightarrow{\alpha} s'_i)$ where

$$s_i = \oplus_{j\in J_i}\ g_{ij}\ (\sigma_{ij};\omega_P;\lambda\widetilde{y}_P.P;\widetilde{v}_{P_{ij}}) \text{ and } s'_i = \oplus_{j\in J'_i}\ g'_{ij}\ (\sigma'_{ij};\omega'_P;\lambda\widetilde{y}'_P.P';\widetilde{v}'_{P_{ij}})\ .$$

For each $i \in I$, because $(s_i,t_i) \in \mathcal{R}_1$ there exists t'_i, t''_i such that $t_i \Longrightarrow t'_i \xrightarrow{\alpha} t''_i$ where

$$\begin{aligned} t_i &= \oplus_{k\in K_i}\ h_{ik}\ (\tau_{ik};\omega_Q;\lambda\widetilde{y}_Q.Q;\widetilde{v}_{Q_{ik}}),\\ t'_i &= \oplus_{k\in K'_i}\ h'_{ik}\ (\tau'_{ik};\omega'_Q;\lambda\widetilde{y}'_Q.Q';\widetilde{v}'_{Q_{ik}}) \text{ and}\\ t''_i &= \oplus_{k\in K''_i}\ h''_{ik}\ (\tau''_{ik};\omega''_Q;\lambda\widetilde{y}''_Q.Q'';\widetilde{v}''_{Q_{ik}})\ . \end{aligned}$$

By Lemma 4.12 we have $t \Longrightarrow t' \xrightarrow{\alpha} t''$ where

$$\begin{aligned} t' &= \oplus_{\substack{i\in I\\ k\in K'_i}}\ h'_{ik}\ (\tau'_{ik};\omega'_Q;\lambda\widetilde{y}'_Q.Q';\widetilde{v}'_{Q_{ik}}),\\ t'' &= \oplus_{\substack{i\in I\\ k\in K''_i}}\ h''_{ik}\ (\tau''_{ik};\omega''_Q;\lambda\widetilde{y}''_Q.Q'';\widetilde{v}''_{Q_{ik}}) \end{aligned}$$

and $(s,t') \in \mathcal{R}_2$ and $(s',t'') \in \mathcal{R}_2$.

If $s \xrightarrow{c![U,\widetilde{r}]} s'$ where $s' = \boxplus_{m\in M}\ p_m \bullet s'_m$ and

$$s'_m = \oplus_{\substack{i\in I_m \\ j\in J_{im}}} \frac{f_i}{p_m} g_{ij}\ (\sigma_{ij}; \omega'_P; \lambda\widetilde{y}'_P.P'; \widetilde{v}'_{P_{ij}})$$

then by L-Out we can derive $\forall i \in I.(s_i \xrightarrow{c![U_i,\widetilde{r}]} s'_i)$ where

$$s'_i = \boxplus_{m\in M_i}\ p_{im} \bullet \oplus_{j\in J_{im}} \frac{g_{ij}}{p_{im}}\ (\sigma_{ij}; \omega'_P; \lambda\widetilde{y}'_P.P'; \widetilde{v}'_{P_{ij}})$$

and $U = \bigcup_{i\in I} U_i$ and $M = \bigcup_{i\in I} M_i$ and

$$p_m = \frac{\sum_{i\in I_m} p_{im}}{\sum_{i\in I} p_{im}}\ .$$

For each $i \in I$, because $(s_i, t_i) \in \mathcal{R}_1$ there exist t'_i, t''_i such that $t_i \Longrightarrow t'_i \xrightarrow{c![U_i,\widetilde{r}]} t''_i$. Using L-Out we can derive the transitions $t \Longrightarrow t' \xrightarrow{c![U,\widetilde{r}]} t''$ where $t'' = \boxplus_{m\in M}\ p_m \bullet t''_m$ and $(s, t') \in \mathcal{R}_2$ and $(s', t'') \in \mathcal{R}_3$ and $\forall m \in M.((s'_m, t''_m) \in \mathcal{R}_2)$. □

Lemma 4.19. *Full probabilistic branching bisimilarity is preserved by input prefix, output prefix, action prefix, qubit declaration, channel restriction and non-deterministic choice.*

Proof. Because $P \leftrightarrows^c Q$, there exists a bisimulation $\mathcal{R}_1$ such that for all quantum states σ and for all substitutions $\kappa = \{\widetilde{u}, \widetilde{q}/\widetilde{x}\}$ we have $((\sigma; \widetilde{q}; P\kappa), (\sigma; \widetilde{q}; Q\kappa)) \in \mathcal{R}_1$. For each case we construct a suitable relation and show that it is a probabilistic branching bisimulation. The most complicated cases are for output and action prefix; for these cases we must consider transitions due to L-Out and L-Act respectively, and also transitions derived by L-Expr.

Input prefix: Let $\mathcal{R}_2$ be a relation such that $\forall \sigma, \kappa' = \{\widetilde{v}, \widetilde{r}/\widetilde{y}\}$,

$$((\sigma; \widetilde{r}; c?[\widetilde{x}].P\kappa'), (\sigma; \widetilde{r}; c?[\widetilde{x}].Q\kappa')) \in \mathcal{R}_2\ .$$

We now show that $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ is a bisimulation: There is only one transition possibly, namely an input action. If $(\sigma; \widetilde{r}; c?[\widetilde{x}].P\kappa') \xrightarrow{c?[\widetilde{u},\widetilde{q}]} (\sigma; \widetilde{r}, \widetilde{q}; P\kappa'\kappa) = s'$ then we also have $(\sigma; \widetilde{r}; c?[\widetilde{x}].Q\kappa') \xrightarrow{c?[\widetilde{u},\widetilde{q}]} (\sigma; \widetilde{r}, \widetilde{q}; Q\kappa'\kappa) = t'$ and $(s', t') \in \mathcal{R}_1$.

Output prefix: Define an equivalence relation $\mathcal{R}_2$ such that for all σ, κ,

$$((\sigma; \widetilde{q}; c![\widetilde{e}].P\kappa), (\sigma; \widetilde{q}; c![\widetilde{e}].Q\kappa)) \in \mathcal{R}_2$$

whenever $P \leftrightarrows Q$. Then define $\mathcal{R}$ as the relation

$$\begin{aligned}\mathcal{R} = \{(&\uplus_{m\in M}\ p_m \bullet \oplus_{i\in I_m}\ g_i\ (\sigma_{im}; \widetilde{q}; \lambda\widetilde{x}.P\kappa; \widetilde{v}_{im}),\\ &\uplus_{m\in M}\ p_m \bullet \oplus_{i\in I_m}\ g_i\ (\sigma_{im}; \widetilde{q}; \lambda\widetilde{x}.Q\kappa; \widetilde{v}_{im}))\\ &\mid \forall m \in M, i \in I_m.((\sigma_{im}; \widetilde{q}; P\kappa\kappa'), (\sigma_{im}; \emptyset; Q\kappa\kappa')) \in \mathcal{R}_1 \cup \mathcal{R}_2\}\ .\end{aligned}$$

where $\kappa' = \{\widetilde{v}_{im}/\widetilde{x}\}$. Note that we also include non-probabilistic configurations in $\mathcal{R}$; these correspond to the cases when M is a singleton set. The possible transitions are ultimately derived by either R-Plus, R-Measure, R-Trans or L-Out; we consider each case in turn.

R-Plus: Let

$$\begin{aligned}s &= \oplus_{i\in I}\ g_i\ (\sigma_i; \widetilde{q}; \lambda\widetilde{x}.c![\widetilde{e}].P\kappa; \widetilde{v}_i) \text{ and}\\ s' &= \oplus_{i\in I}\ g_i\ (\sigma_i; \widetilde{q}; \lambda\widetilde{x}y.c![\widetilde{e}'].P\kappa; \widetilde{v}_i, u_i)\ .\end{aligned}$$

If $s \xrightarrow{\tau} s'$ then $t \xrightarrow{\tau} t'$ where

$$\begin{aligned}t &= \oplus_{i\in I}\ g_i\ (\sigma_i; \widetilde{q}; \lambda\widetilde{x}.c![\widetilde{e}].Q\kappa; \widetilde{v}_i) \text{ and}\\ t' &= \oplus_{i\in I}\ g_i\ (\sigma_i; \widetilde{q}; \lambda\widetilde{x}y.c![\widetilde{e}'].Q\kappa; \widetilde{v}_i, u_i)\ .\end{aligned}$$

We have $\forall i \in I$, $((\sigma_i; \widetilde{q}; c![\widetilde{e}']\{\widetilde{v}_i u_i/\widetilde{x}y\}P\kappa), (\sigma_i; \widetilde{q}; c![\widetilde{e}']\{\widetilde{v}_i u_i/\widetilde{x}y\}Q\kappa)) \in \mathcal{R}_2$, therefore $(s', t') \in \mathcal{R}$.

R-Measure: Let

$$\begin{aligned}s &= \oplus_{i\in I}\ g_i\ (\sigma_i; \widetilde{q}; \lambda\widetilde{x}.c![\widetilde{e}].P\kappa; \widetilde{v}_i) \text{ and}\\ s' &= \oplus_{\substack{i\in I\\ j\in J_i}}\ g_i h_{ij}\ (\sigma_{ij}; \widetilde{q}; \lambda\widetilde{x}\widetilde{y}.c![\widetilde{e}'].P\kappa; \widetilde{v}_i, \widetilde{v}_{ij})\ .\end{aligned}$$

If $s \xrightarrow{\tau} s'$ then $t \xrightarrow{\tau} t'$ where

$$\begin{aligned}t &= \oplus_{i\in I}\ g_i\ (\sigma_i; \widetilde{q}; \lambda\widetilde{x}.c![\widetilde{e}].Q\kappa; \widetilde{v}_i) \text{ and}\\ t' &= \oplus_{\substack{i\in I\\ j\in J_i}}\ g_i h_{ij}\ (\sigma_{ij}; \widetilde{q}; \lambda\widetilde{x}\widetilde{y}.c![\widetilde{e}'].Q\kappa; \widetilde{v}_i, \widetilde{v}_{ij})\ .\end{aligned}$$

We have $\forall i \in I, j \in J_i$, $((\sigma_i; \widetilde{q}; c![\widetilde{e}']\{\widetilde{v}_i\widetilde{v}_{ij}/\widetilde{x}\widetilde{y}\}.P\kappa), (\sigma_i; \widetilde{q}; c![\widetilde{e}']\{\widetilde{v}_i\widetilde{v}_{ij}/\widetilde{x}\widetilde{y}\}.Q\kappa)) \in \mathcal{R}_2$, therefore $(s', t') \in \mathcal{R}$.

R-Trans: Let

$$\begin{aligned}s &= \oplus_{i\in I}\ g_i\ (\sigma_i; \widetilde{q}; \lambda\widetilde{x}.c![\widetilde{e}].P\kappa; \widetilde{v}_i) \text{ and}\\ s' &= \oplus_{i\in I}\ g_i\ (\sigma'_i; \widetilde{q}; \lambda\widetilde{x}.c![\widetilde{e}'].P\kappa; \widetilde{v}_i)\ .\end{aligned}$$

If $s \xrightarrow{\tau} s'$ then $t \xrightarrow{\tau} t'$ where

$$t = \oplus_{i \in I}\ g_i\ (\sigma_i; \widetilde{q}; \lambda \widetilde{x}.c![\widetilde{e}].Q\kappa; \widetilde{v}_i) \text{ and}$$
$$t' = \oplus_{i \in I}\ g_i\ (\sigma'_i; \widetilde{q}; \lambda \widetilde{x}y.c![\widetilde{e'}].Q\kappa; \widetilde{v}_i)\ .$$

We have for all $i \in I$, $((\sigma'_i; \widetilde{q}; c![\widetilde{e'}]\{\widetilde{v}_i/\widetilde{x}\}P\kappa), (\sigma'_i; \widetilde{q}; c![\widetilde{e'}]\{\widetilde{v}_i/\widetilde{x}\}Q\kappa)) \in \mathcal{R}_2$, therefore $(s', t') \in \mathcal{R}$.

L-Out: If

$$\oplus_i\ g_i\ (\sigma_i; \widetilde{q}; \lambda \widetilde{x}\widetilde{y}.c![\widetilde{x}, \widetilde{r}].P\kappa; \widetilde{u}_i, \widetilde{v}_i) \xrightarrow{c![U,\widetilde{r}]} \boxplus_{m \in M}\ p_m \bullet s'_m$$

where $s'_m = \oplus_{i \in I_m}\ \frac{g_i}{p_m}\ (\sigma_i; \widetilde{q}'; \lambda \widetilde{x}\widetilde{y}.P\kappa; \widetilde{u}_i, \widetilde{v}_i)$ and $\widetilde{q} = \widetilde{q}'\widetilde{r}$ and $U = \{\widetilde{u}_i\}$ then

$$\oplus_i\ g_i\ (\sigma_i; \widetilde{q}; \lambda \widetilde{x}\widetilde{y}.c![\widetilde{x}, \widetilde{r}].Q\kappa; \widetilde{u}_i, \widetilde{v}_i) \xrightarrow{c![U,\widetilde{r}]} \boxplus_{m \in M}\ p_m \bullet t'_m$$

where $t'_m = \oplus_{i \in I_m}\ \frac{g_i}{p_m}\ (\sigma_i; \widetilde{q}'; \lambda \widetilde{x}\widetilde{y}.Q\kappa; \widetilde{u}_i, \widetilde{v}_i)$. By IT-Out and Lemma 3.10 we have $\widetilde{r} \notin fq(P\kappa)$ and $\widetilde{r} \notin fq(Q\kappa)$, therefore $P\kappa = P\kappa'$ and $Q\kappa = Q\kappa'$ where $\kappa' = \{\widetilde{u}\widetilde{q}'/\widetilde{x}\}$. Then we have $\forall m \in M, i \in I_m$

$$((\sigma_i; \widetilde{q}'; P\kappa'\{\widetilde{u}_i\widetilde{v}_i/\widetilde{x}\widetilde{y}\}), (\sigma_i; \widetilde{q}'; Q\kappa'\{\widetilde{u}_i\widetilde{v}_i/\widetilde{x}\widetilde{y}\})) \in \mathcal{R}_1,$$

therefore $\forall m \in M.(s'_m, t'_m) \in \mathcal{R}$.

Qubit declaration: Define a relation

$$\mathcal{R}_2 = \{((\sigma; \widetilde{q}; (\mathsf{qbit}\ x).P\kappa), (\sigma; \widetilde{q}; (\mathsf{qbit}\ x).Q\kappa)) \mid ((\sigma; \widetilde{q}; P\kappa), (\sigma; \widetilde{q}; Q\kappa)) \in \mathcal{R}_1\}\ .$$

Then $(\sigma; \widetilde{q}; (\mathsf{qbit}\ x).P\kappa) \xrightarrow{\tau} (\sigma'; \widetilde{q}, r; P\kappa\kappa')$ where $\kappa' = \{r/x\}$ and r is fresh. We also have $(\sigma; \widetilde{q}; (\mathsf{qbit}\ x).Q\kappa) \xrightarrow{\tau} (\sigma'; \widetilde{q}, r; Q\kappa\kappa')$. Then $((\sigma'; \widetilde{q}, r; P\kappa\kappa'), (\sigma'; \widetilde{q}, r; Q\kappa\kappa')) \in \mathcal{R}_1$, hence $\mathcal{R}_1 \cup \mathcal{R}_2$ is a bisimulation.

Restriction: Given a configuration $s = \oplus_i\ g_i\ (\sigma_i; \omega_1; \lambda \widetilde{x}.P; \widetilde{u}_i)$, let s_n denote the corresponding configuration with a restriction $\oplus_i\ g_i\ (\sigma_i; \omega_1; \lambda \widetilde{x}.(\nu c)P; \widetilde{u}_i)$. Define a relation

$$\mathcal{R}_2 = \{(s_n, t_n) \mid (s, t) \in \mathcal{R}_1\}\ .$$

If $s_n \xrightarrow{\alpha} s'_n$ then by L-Res we have $s \xrightarrow{\alpha} s'$. Because $(s, t) \in \mathcal{R}_1$, there exist t', t'' such that $t \Longrightarrow t' \xrightarrow{\alpha} t''$ and $(s, t') \in \mathcal{R}_1$ and $(s', t'') \in \mathcal{R}_1$. By L-Res we have $t_n \Longrightarrow t'_n \xrightarrow{\alpha} t''_n$ and we have $(s_n, t'_n) \in \mathcal{R}_2$ and $(s'_n, t''_n) \in \mathcal{R}_2$.

Action prefix: Define a relation $\mathcal{R}_2$ as

$$\mathcal{R}_2 = \{((\sigma; \widetilde{q}; \{e\}.P\kappa), (\sigma; \widetilde{q}; \{e\}.Q\kappa)) \mid ((\sigma; \widetilde{q}; P\kappa), (\sigma; \widetilde{q}; Q\kappa)) \in \mathcal{R}_1\}\ .$$

Then define

$$\mathcal{R}_3 = \{(\oplus_i\ g_i\ (\sigma_i;\widetilde{q};\lambda\widetilde{y}.\{e\}.P\kappa;\widetilde{v}_i), \oplus_i\ g_i\ (\sigma_i;\widetilde{q};\lambda\widetilde{y}.\{e\}.Q\kappa;\widetilde{v}_i)) \mid \forall i.((\sigma_i;\widetilde{q};\{e\}\{\widetilde{v}_i/\widetilde{y}\}.P\kappa),(\sigma_i;\widetilde{q};\{e\}\{\widetilde{v}_i/\widetilde{y}\}.Q\kappa)) \in \mathcal{R}_2\}\ .$$

Then, for $(s,t) \in \mathcal{R}_3$, if $s \xrightarrow{\tau} s'$ where $s' = \oplus_{ij}\ g_i h_{ij}\ (\sigma_{ij};\widetilde{q};\lambda\widetilde{y}\widetilde{y}'.\{e'\}.P\kappa;\widetilde{v}_i,\widetilde{v}_{ij})$ then $t \xrightarrow{\tau} t'$ where $t' = \oplus_{ij}\ g_i h_{ij}\ (\sigma_{ij};\widetilde{q};\lambda\widetilde{y}\widetilde{y}'.\{e'\}.Q\kappa;\widetilde{v}_i,\widetilde{v}_{ij})$ and for each i,j we have $((\sigma_{ij};\widetilde{q};\{e'\}\{\widetilde{v}_i\widetilde{v}_{ij}/\widetilde{y}\widetilde{y}'\}.P\kappa),(\sigma_{ij};\widetilde{q};\{e'\}\{\widetilde{v}_i\widetilde{v}_{ij}/\widetilde{y}\widetilde{y}'\}.P\kappa)) \in \mathcal{R}_2$ therefore $(s',t') \in \mathcal{R}_3$.

If $s \xrightarrow{\tau} s'$ by L-Act where $s' = \oplus_i\ g_i\ (\sigma_i;\widetilde{q};P\kappa)$ since variables $\widetilde{y}$ are not in $P\kappa$, then $t \xrightarrow{\tau} t'$ where $t' = \oplus_i\ g_i\ (\sigma_i;\widetilde{q};Q\kappa)$. By Lemma 4.18 we have $s' \leftrightarrows t'$.

Non-deterministic choice: Because $R \leftrightarrows^c R$ there exists a bisimulation $\mathcal{R}_2$ such that $\forall\sigma,\kappa.((\sigma;\widetilde{q};R),(\sigma;\widetilde{q};R)) \in \mathcal{R}_2$, and because $\alpha.P \leftrightarrows^c \alpha.Q$ from the previous cases, there is a bisimulation $\mathcal{R}_3$ such that $\forall\sigma,\kappa.((\sigma;\widetilde{q};\alpha.P\kappa),(\sigma;\widetilde{q};\alpha.Q\kappa)) \in \mathcal{R}_3$. Now define a relation $\mathcal{R}_4$ such that

$$\mathcal{R}_4 = \{((\sigma;\widetilde{q};\alpha.P\kappa + R),(\sigma;\widetilde{q};\alpha.Q\kappa + R)) \mid P \leftrightarrows^c Q\}\ .$$

If we have the derivation

$$\frac{(\sigma;\widetilde{q};\alpha.P\kappa) \xrightarrow{\beta} s'}{(\sigma;\widetilde{q};\alpha.P\kappa + R) \xrightarrow{\beta} s'}$$

then $(\sigma;\widetilde{q};\alpha.Q\kappa) \xrightarrow{\tau} t'$ and $(s',t') \in \mathcal{R}_3$. Therefore by L-Sum we have the transition $(\sigma;\widetilde{q};\alpha.Q\kappa + R) \xrightarrow{\beta} t'$. Note that the prefix α guarantees that this transition is strongly matched.

If $(\sigma;\widetilde{q};\alpha.P\kappa + R) \xrightarrow{\beta} s''$ is derived from the transition $(\sigma;\widetilde{q};R) \xrightarrow{\beta} s''$ then by L-Sum we have $(\sigma;\widetilde{q};\alpha.Q\kappa + R) \xrightarrow{\beta} t''$ where $s'' = t''$, hence $(s'',t'') \in \mathcal{R}_2$. Therefore $\mathcal{R}_2 \cup \mathcal{R}_3 \cup \mathcal{R}_4$ is a bisimulation. □

We have now proved that full probabilistic branching bisimilarity is preserved by all process constructs, hence we can state and prove the following theorem.

Theorem 4.20 (Full probabilistic branching bisimilarity is a congruence)**.** *If $P \leftrightarrows^c Q$ then for any context C, if $\Gamma \vdash C[P]$ and $\Gamma \vdash C[Q]$ then $C[P] \leftrightarrows^c C[Q]$.*

Proof. Follows directly from Theorem 4.15 and Lemma 4.19. □

4.4 Applications

In this section, we apply full probabilistic branching bisimilarity to some quantum protocols. In particular, we begin by reconsidering the equivalence of teleportation and a quantum channel that was presented in the previous chapter.

$([\widetilde{r}p \mapsto |\phi_0\rangle|0\rangle + |\phi_1\rangle|1\rangle]; \emptyset; \mathit{Teleport})$
$\stackrel{\tau}{\Longrightarrow}$ (L-Qbit, R-Trans, L-Act, R-Trans, L-Act)
$([\widetilde{r}pq_1q_2 \mapsto (|\phi_0\rangle|0\rangle + |\phi_1\rangle|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; q_1, q_2; (\nu e)(\mathit{Alice}\{q_2/z\} \parallel \mathit{Bob}\{q_1/y\}))$
$\stackrel{c?[p]}{\longrightarrow}$ (L-In)
$([\widetilde{r}pq_1q_2 \mapsto (|\phi_0\rangle|0\rangle + |\phi_1\rangle|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; q_1, q_2, p;$
$\quad (\nu e)(\{p, q_2 *= \mathsf{CNot}\}.\{p *= \mathsf{H}\}.e![\mathsf{measure}\ q_2.\mathsf{measure}\ p].\mathbf{0} \parallel \mathit{Bob}\{q_1/y\}))$
$\stackrel{\tau}{\Longrightarrow}$ (R-Trans, L-Act, R-Trans, L-Act)
$([\widetilde{r}pq_1q_2 \mapsto \frac{1}{2}|\phi_0\rangle(|000\rangle + |100\rangle + |011\rangle + |111\rangle) + \frac{1}{2}|\phi_1\rangle(|001\rangle - |101\rangle + |010\rangle - |110\rangle)];$
$\quad q_1, q_2, p; (\nu e)(e![\mathsf{measure}\ q_2, \mathsf{measure}\ p].\mathbf{0} \parallel \mathit{Bob}\{q_1/y\}))$
$\stackrel{\tau}{\Longrightarrow}$ (R-Measure, R-Measure)
$\oplus_{\substack{i\in\{0,1\}\\ j\in\{0,1\}}} \frac{1}{4}\ ([\widetilde{r}pq_1q_2 \mapsto |\psi_{ij}\rangle]; q_1, q_2, p; \lambda uv.(\nu e)(e![u, v].\mathbf{0} \parallel \mathit{Bob}\{q_1/y\}); i, j)$
$\stackrel{\tau}{\Longrightarrow}$ (L-Com, R-Trans, L-Act, R-Trans, L-Act)
$\oplus_{\substack{i\in\{0,1\}\\ j\in\{0,1\}}} \frac{1}{4}\ ([\widetilde{r}pq_1q_2 \mapsto |\psi'_{ij}\rangle]; q_1, q_2, p; \lambda uv.(\nu e)(d![q_1].\mathbf{0}); i, j)$
$\stackrel{d![q_1]}{\longrightarrow}$ (L-Out)
$\oplus_{\substack{i\in\{0,1\}\\ j\in\{0,1\}}} \frac{1}{4}\ ([\widetilde{r}pq_1q_2 \mapsto |\psi'_{ij}\rangle]; q_1, q_2, p; \lambda uv.\mathbf{0}; i, j)$

where

$$
\begin{aligned}
|\psi_{00}\rangle &= |\phi_0\rangle|000\rangle + |\phi_1\rangle|010\rangle \\
|\psi_{01}\rangle &= |\phi_0\rangle|100\rangle - |\phi_1\rangle|110\rangle \\
|\psi_{10}\rangle &= |\phi_0\rangle|011\rangle + |\phi_1\rangle|001\rangle \\
|\psi_{11}\rangle &= |\phi_0\rangle|111\rangle - |\phi_1\rangle|101\rangle \\
|\psi'_{00}\rangle &= |\phi_0\rangle|000\rangle + |\phi_1\rangle|010\rangle \\
|\psi'_{01}\rangle &= |\phi_0\rangle|100\rangle + |\phi_1\rangle|110\rangle \\
|\psi'_{10}\rangle &= |\phi_0\rangle|001\rangle + |\phi_1\rangle|011\rangle \\
|\psi'_{11}\rangle &= |\phi_0\rangle|101\rangle + |\phi_1\rangle|111\rangle
\end{aligned}
$$

Figure 4.4. Execution of quantum teleportation.

4.4.1 Quantum Teleportation

The CQP model of teleportation was defined in Section 3.3 (Figure 3.8). With the introduction of mixed configurations, the execution of the protocol is different. Figure 4.4 shows the execution of *Teleport* with respect to the new semantics. The creation of mixed configurations due to the two measurements can be seen, however probabilistic branching never occurs because there is no observable output that can distinguish the components.

Lemma 4.21. *QChannel* $\leftrightarrows$ *Teleport*.

Proof. We follow a similar argument to the proof of Lemma 3.18, however we do not need to consider probabilistic branching in this scenario. We construct an equivalence

relation $\mathcal{R}$ that contains the pair $((\sigma;\emptyset;\mathit{Teleport}),(\sigma;\emptyset;\mathit{QChannel}))$ for all σ and is closed under their transitions. Let

$$S_1(\sigma) = \{s \mid (\sigma;\emptyset;P) \Longrightarrow s, P \in \{\mathit{Teleport}, \mathit{QChannel}\}\}$$
$$S_2(\sigma) = \{s \mid (\sigma;\emptyset;P) \stackrel{c?[p]}{\Longrightarrow} s, P \in \{\mathit{Teleport}, \mathit{QChannel}\}\}$$
$$S_3(\sigma) = \{s \mid (\sigma;\emptyset;P) \stackrel{c?[p]}{\Longrightarrow}\stackrel{d![q]}{\Longrightarrow} s, P \in \{\mathit{Teleport}, \mathit{QChannel}\}\} \ .$$

Then define $\mathcal{R}$ to be the relation where $S_1(\sigma), S_2(\sigma)$ and $S_3(\sigma)$ are the equivalence classes:

$$\mathcal{R} = \bigcup_{i\in\{1,2,3\}}^{\sigma} \{(s,t) \mid s,t \in S_i(\sigma)\} \ .$$

We now prove that $\mathcal{R}$ is a bisimulation.

If $s,t \in S_1(\sigma)$ and if $s \xrightarrow{\tau} s'$ then we have $s' \in S_1(\sigma)$ and therefore $(s',t) \in \mathcal{R}$. Otherwise if $s \xrightarrow{c?[p]} s'$ then $s' \in S_2(\sigma)$ and we find t', t'' such that $t \Longrightarrow t' \xrightarrow{c?[p]} t''$ with $t' \in S_1(\sigma)$ and $t'' \in S_2(\sigma)$, therefore $(s,t') \in \mathcal{R}$ and $(s',t'') \in \mathcal{R}$.

If $s,t \in S_2(\sigma)$ and if $s \xrightarrow{\tau} s'$ then we have $s' \in S_2(\sigma)$ and therefore $(s',t) \in \mathcal{R}$. Otherwise if $s \xrightarrow{d![r_1]} s'$ then $s' \in S_3(\sigma)$ and we find t', t'' such that $t \Longrightarrow t' \xrightarrow{d![r_2]} t''$ with $t' \in S_2(\sigma)$ and $t'' \in S_3(\sigma)$. If s is a mixed configuration arising from *Teleport* then for an arbitrary state $\sigma = [\widetilde{r}p \mapsto |\phi_0\rangle|0\rangle + |\phi_1\rangle|1\rangle]$, with reference to Figure 4.4, $\rho^{q_1\widetilde{r}}(s) = \frac{1}{4}\rho^{q_1\widetilde{r}}(|\psi_{00}\rangle) + \frac{1}{4}\rho^{q_1\widetilde{r}}(|\psi_{01}\rangle) + \frac{1}{4}\rho^{q_1}(\widetilde{r}|\psi_{10}\rangle) + \frac{1}{4}\rho^{q_1\widetilde{r}}(|\psi_{11}\rangle) = \sum_{i\in\{0,1\},j\in\{0,1\}} |i\rangle\langle j||\phi_i\rangle\langle\phi_j|$. Otherwise if s arises from *QChannel* then we also find that $\rho^{q_1\widetilde{r}}(s) = \sum_{i\in\{0,1\},j\in\{0,1\}} |i\rangle\langle j||\phi_i\rangle\langle\phi_j|$. The same reasoning applies to t', hence we have $\rho^{q_1\widetilde{r}}(s) = \rho^{q_2\widetilde{r}}(t')$.

If $s,t \in S_3(\sigma)$ then there are no possible transitions. □

We can extend this result to full probabilistic branching bisimilarity; first we need the following result.

Lemma 4.22 (Weakening for qubit list)**.** *If* $(\sigma_1;\omega_1;P) \leftrightarrows (\sigma_2;\omega_2;Q)$ *and* $\widetilde{r} \cap \omega_1 = \widetilde{r} \cap \omega_2 = \emptyset$*, then* $(\sigma_1;\omega_1,\widetilde{r};P) \leftrightarrows (\sigma_2;\omega_2,\widetilde{r};Q)$.

Proof. Define a relation

$$\mathcal{R} = \{((\sigma_1;\omega_1,\widetilde{r};P),(\sigma_2;\omega_2,\widetilde{r};Q)) \mid (\sigma_1;\omega_1;P) \leftrightarrows (\sigma_2;\omega_2;Q) \text{ and } \widetilde{r}\cap\omega_1 = \widetilde{r}\cap\omega_2 = \emptyset\}$$

Then $\mathcal{R}$ is a probabilistic branching bisimulation. □

Lemma 4.23. *Teleport* $\leftrightarrows^c$ *QChannel*.

Proof. Because *Teleport* $\leftrightarrows$ *QChannel*, there is a probabilistic branching bisimulation $\mathcal{R}$ such that $((\sigma;\emptyset;\mathit{Teleport}),(\sigma;\emptyset;\mathit{QChannel})) \in \mathcal{R}$ for all σ. Both *Teleport* and

$$
\begin{aligned}
\mathit{DenseC} &= (\mathsf{qbit}\ q_1, q_2)\{q_1 *\!= \mathsf{H}\}.\{q_1, q_2 *\!= \mathsf{CNot}\}.(\nu e{:}\widehat{\ }[\mathsf{Qbit}])(\mathit{Alice}(q_1) \parallel \mathit{Bob}(q_2)) \\
\mathit{Alice}(q_1) &= c?[a{:}\mathsf{Bit}, b{:}\mathsf{Bit}].\{q_1 *\!= \mathsf{X}^b\}.\{q_1 *\!= \mathsf{Z}^a\}.e![q_1].\mathbf{0} \\
\mathit{Bob}(q_2) &= e?[q_1{:}\mathsf{Qbit}].\{q_1, q_2 *\!= \mathsf{CNot}\}.\{q_1 *\!= \mathsf{H}\}.d![\mathsf{measure}\ q_1, \mathsf{measure}\ q_2].\mathbf{0} \\
\mathit{CChannel} &= c?[a{:}\mathsf{Bit}, b{:}\mathsf{Bit}].d![a, b].\mathbf{0}
\end{aligned}
$$

Figure 4.5. CQP model for superdense coding and its specification.

QChannel have no free variables, therefore for any substitution $\kappa = \{\widetilde{u}\widetilde{r}/\widetilde{x}\}$ we have *Teleport* κ = *Teleport* and *QChannel* κ = *QChannel*. Therefore for all σ and for all κ we have
$((\sigma; \emptyset;$ *Teleport* $\kappa), (\sigma; \emptyset;$ *QChannel* $\kappa)) \in \mathcal{R}$. Then by Lemma 4.22 we have
$(\sigma; \widetilde{r};$ *Teleport* $\kappa) \leftrightarrows (\sigma; \widetilde{r};$ *QChannel* $\kappa)$ □

It follows from Theorem 4.20 that *Teleport* and *QChannel* are congruent processes.

Corollary 4.24. *Teleport* = *QChannel*

4.4.2 Superdense Coding

The superdense coding protocol was described in Section 2.2.2. Figure 4.5 presents a CQP model for superdense coding, *DenseC*. This CQP model, unlike the circuit model, is able to clearly describe the actions of the two users using the processes *Alice* and *Bob*.

We have also described a high-level specification for superdense coding, *CChannel*. Essentially, this specification is a 2-bit classical channel. We shall now show that *DenseC* is bisimilar to *CChannel*. First, we shall describe the execution of *DenseC* and then we shall formally define an equivalence relation. Based on the execution, we can argue that this relation is a probabilistic branching bisimulation.

Consider an arbitrary quantum state $[\widetilde{r} \mapsto |\psi\rangle]$. Let $s = ([\widetilde{r} \mapsto |\psi\rangle]; \emptyset;$ *DenseC*),

then the execution is as follows.

$$
\begin{aligned}
s &\xrightarrow{\tau}([\widetilde{r}q_1q_2 \mapsto |\psi_1\rangle]; q_1, q_2; \{q_1 *= \mathsf{H}\}.\{q_1, q_2 *= \mathsf{CNot}\}.(\nu e{:}\widehat{}[\mathsf{Qbit}])(\mathit{Alice}(q_1) \parallel \mathit{Bob}(q_2)))\\
&\xrightarrow{\tau}([\widetilde{r}q_1q_2 \mapsto |\psi_2\rangle]; q_1, q_2; \{q_1, q_2 *= \mathsf{CNot}\}.(\nu e{:}\widehat{}[\mathsf{Qbit}])(\mathit{Alice}(q_1) \parallel \mathit{Bob}(q_2)))\\
&\xrightarrow{\tau}([\widetilde{r}q_1q_2 \mapsto |\psi_3\rangle]; q_1, q_2; (\nu e{:}\widehat{}[\mathsf{Qbit}])(\mathit{Alice}(q_1) \parallel \mathit{Bob}(q_2)))\\
&\xrightarrow{c?[\alpha,\beta]}([\widetilde{r}q_1q_2 \mapsto |\psi_4\rangle]; q_1, q_2; (\nu e{:}\widehat{}[\mathsf{Qbit}])(\{q_1 *= \mathsf{X}^b\}.\{q_1 *= \mathsf{Z}^a\}.e![q_1].\mathbf{0} \parallel \mathit{Bob}(q_2)))\\
&\xrightarrow{\tau}([\widetilde{r}q_1q_2 \mapsto |\psi_5\rangle]; q_1, q_2; (\nu e{:}\widehat{}[\mathsf{Qbit}])(\{q_1 *= \mathsf{Z}^a\}.e![q_1].\mathbf{0} \parallel \mathit{Bob}(q_2)))\\
&\xrightarrow{\tau}([\widetilde{r}q_1q_2 \mapsto |\psi_6\rangle]; q_1, q_2; (\nu e{:}\widehat{}[\mathsf{Qbit}])(e![q_1].\mathbf{0} \parallel \mathit{Bob}(q_2)))\\
&\xrightarrow{\tau}([\widetilde{r}q_1q_2 \mapsto |\psi_7\rangle]; q_1, q_2; (\nu e{:}\widehat{}[\mathsf{Qbit}])(\{q_1, q_2 *= \mathsf{CNot}\}.\{q_1 *= \mathsf{H}\}.\\
&\qquad\qquad d![\mathsf{measure}\ q_1, \mathsf{measure}\ q_2].\mathbf{0}))\\
&\xrightarrow{\tau}([\widetilde{r}q_1q_2 \mapsto |\psi_8\rangle]; q_1, q_2; (\nu e{:}\widehat{}[\mathsf{Qbit}])(\{q_1 *= \mathsf{H}\}.d![\mathsf{measure}\ q_1, \mathsf{measure}\ q_2].\mathbf{0}))\\
&\xrightarrow{\tau}([\widetilde{r}q_1q_2 \mapsto |\psi_9\rangle]; q_1, q_2; (\nu e{:}\widehat{}[\mathsf{Qbit}])(d![\mathsf{measure}\ q_1, \mathsf{measure}\ q_2].\mathbf{0}))\\
&\xrightarrow{\tau}([\widetilde{r}q_1q_2 \mapsto |\psi_{10}\rangle]; q_1, q_2; (\nu e{:}\widehat{}[\mathsf{Qbit}])(d![\alpha, \mathsf{measure}\ q_2].\mathbf{0}))\\
&\xrightarrow{\tau}([\widetilde{r}q_1q_2 \mapsto |\psi_{10}\rangle]; q_1, q_2; (\nu e{:}\widehat{}[\mathsf{Qbit}])(d![\alpha, \beta].\mathbf{0}))\\
&\xrightarrow{d![\alpha,\beta]}([\widetilde{r}q_1q_2 \mapsto |\psi_{10}\rangle]; q_1, q_2; \mathbf{0})
\end{aligned}
$$

The quantum states that arise in the execution are clearly dependent on the two bit values α and β that are received. This results in four possible executions – it is important to note that these are neither probabilistic branches nor components in a mixed configuration.

Table 4.1 shows the quantum states and measurement outcomes for each of the four possible values that α, β can take. The normalisation factors and the state $|\psi\rangle$ have been omitted for convenience.

It is interesting to note that the measurements in this protocol do not result in mixed configurations. This is an example of a protocol in which the measurements are designed to reveal specific information about the quantum state.

Let $t = ([\widetilde{r} \mapsto |\psi\rangle]; \emptyset; \mathit{CChannel})$. Then define an equivalence relation $\mathcal{R}$ by the equivalence classes C_1, $C_2(\alpha, \beta)$ and C_3 as follows.

$(u', v') \in C_1$ if $u \Longrightarrow u'$ and $v \Longrightarrow v'$ for $u, v \in \{s, t\}$,

$(u', v') \in C_2(\alpha, \beta)$ if $u' \overset{c?[\alpha,\beta]}{\Longrightarrow} u'$ and $v \overset{c?[\alpha,\beta]}{\Longrightarrow} v'$ for $u, v \in \{s, t\}$,

$(u', v') \in C_3$ if $u \overset{c?[\alpha,\beta]d![\alpha,\beta]}{\Longrightarrow} u'$ and $v \overset{c?[\alpha,\beta]d![\alpha,\beta]}{\Longrightarrow} u'$ $\forall \alpha, \beta \in \{0, 1\}$ and $u, v \in \{s, t\}$

The class $C_2(\alpha, \beta)$ is parameterised by the bit values, resulting in 4 distinct classes, while the class C_3 only requires the input and output values to be identical.

Lemma 4.25. *DenseC* $\leftrightarrows$ *CChannel*.

α	0	1	0	1
β	0	0	1	1
$\|\psi_1\rangle$	$\|00\rangle$			
$\|\psi_2\rangle$	$\|00\rangle + \|10\rangle$			
$\|\psi_3\rangle$	$\|00\rangle + \|11\rangle$			
$\|\psi_4\rangle$	$\|00\rangle + \|11\rangle$			
$\|\psi_5\rangle$	$\|00\rangle + \|11\rangle$		$\|10\rangle + \|01\rangle$	
$\|\psi_6\rangle$	$\|00\rangle + \|11\rangle$	$\|00\rangle - \|11\rangle$	$\|10\rangle + \|01\rangle$	$\|01\rangle - \|10\rangle$
$\|\psi_7\rangle$	$\|00\rangle + \|11\rangle$	$\|00\rangle - \|11\rangle$	$\|10\rangle + \|01\rangle$	$\|01\rangle - \|10\rangle$
$\|\psi_8\rangle$	$\|00\rangle + \|10\rangle$	$\|00\rangle - \|10\rangle$	$\|11\rangle + \|01\rangle$	$\|01\rangle - \|11\rangle$
$\|\psi_9\rangle$	$\|00\rangle$	$\|10\rangle$	$\|01\rangle$	$\|11\rangle$
$\|\psi_{10}\rangle$	$\|00\rangle$	$\|10\rangle$	$\|01\rangle$	$\|11\rangle$
measure q_1	0	1	0	1
measure q_2	0	0	1	1

Table 4.1. The states and measurement outcomes of the superdense coding protocol for inputs α, β.

Proof. Consider $(u, v) \in C_1$. If $u \xrightarrow{\tau} u'$ then by definition $(u', v) \in C_1$. The only other possible transition is $u \xrightarrow{c?[\alpha,\beta]} u'$ for some $\alpha, \beta \in \{0, 1\}$. We can see that if $s \Longrightarrow v$ then there are configurations v', v'' such that $v \Longrightarrow v' \xrightarrow{c?[\alpha,\beta]} v''$ and by definition $(u, v') \in C_1$ and $(u', v'') \in C_2(\alpha, \beta)$. Alternatively if $v = t$ then $v \xrightarrow{c?[\alpha,\beta]} v'$ where $(u', v') \in C_2(\alpha, \beta)$.

A similar argument applies to $(u, v) \in C_2(\alpha, \beta)$. We note that for $(u, v) \in C_3$, then the process is $\mathbf{0}$.

We have therefore shown that $\mathcal{R}$ is a probabilistic branching bisimulation. Hence, for any quantum state $|\psi'\rangle$ we can define a probabilistic branching bisimulation $\mathcal{R}(|\psi'\rangle)$. The union $\mathcal{R}_u = \bigcup_{|\psi'\rangle} \mathcal{R}(|\psi'\rangle)$ is also a probabilistic branching bisimulation. Thus we have for all σ, $((\sigma; \emptyset; \mathit{DenseC}), (\sigma; \emptyset; \mathit{CChannel})) \in \mathcal{R}_u \subseteq \leftrightarrows$. □

Lemma 4.26. *DenseC* $\leftrightarrows^c$ *CChannel.*

Proof. Lemma 4.25 gives $\forall \sigma.((\sigma; \emptyset; \mathit{DenseC}) \leftrightarrows (\sigma; \emptyset; \mathit{CChannel}))$. Both *DenseC* and *CChannel* have no free variables, therefore for any substitution $\kappa = \{\widetilde{ur}/\widetilde{x}\}$ we have *DenseC* κ = *DenseC* and *CChannel* κ = *CChannel*. Therefore for all σ, κ we have $(\sigma; \emptyset; \mathit{DenseC}\ \kappa) \leftrightarrows (\sigma; \emptyset; \mathit{CChannel}\ \kappa)$. By Lemma 4.22, we have $\forall \sigma, \kappa.((\sigma; \widetilde{r}; \mathit{DenseC}\ \kappa) \leftrightarrows (\sigma; \widetilde{r}; \mathit{CChannel}\ \kappa))$. Therefore *DenseC* $\leftrightarrows^c$ *CChannel*. □

Corollary 4.27. *DenseC* = *CChannel.*

4.5 Discussion

In this section, we discuss the semantic model that has been introduced in this chapter, and we consider some issues that might arise with potential extensions to the language.

The aim of the new semantics is to capture the observational properties of both the quantum and classical states. The density matrix formalism is used to describe quantum subsystems, and therefore represents the information a particular agent (or process) can determine from his share of the quantum state (the qubits owned by that process). This description is affected by the classical information that arises from measurement outcomes, and it is this link that was not respected in Chapter 3.

Mixed quantum states are ensembles of pure states that combine classical probabilities with quantum states. We have replaced the probabilistic branching behaviour of measurements by incorporating the probabilistic information into mixed states, however, this change is not a straightforward one. The classical values of measurement outcomes are important in the process calculus setting, hence our use of *mixed configurations* has arisen from the need to represent probabilistic ensembles of quantum states alongside their respective classical values.

The crucial part of this approach, is to combine mixed configurations with probabilistic branching. From an observational point of view, the probabilistic distribution of configurations will change if information concerning measurement outcomes is output. Up until the point of output, this classical information is internal to the process, but afterwards it becomes global information. It is this globalisation that coincides with branching.

It is worth considering whether it matters that probabilistic branching is performed at some point after a measurement, even though it is the measurement operation that produces a probabilistic outcome. The subtlety here is in the distinction of quantum and classical uncertainty; the act of measurement changes the former into the latter, while probabilistic branching resolves the resulting classical uncertainty. The branching itself describes the transition from a state in which the observer has uncertainty to a state in which this is resolved, while the associated probabilities quantify the uncertainty.

Labelled transitions vs. Reductions. The labelled transition semantics defined in Chapter 3 are closely linked to the original reduction semantics in [Gay and Nagarajan 2005], providing the additional input and output actions to describe external interactions. In this chapter, the semantics was radically changed to coincide with the observational properties of quantum information. We briefly consider the validity of the reduction semantics in light of these findings, and whether it is compatible with the new semantics.

The reduction semantics is designed to model systems that are closed, and there-

fore no input or output actions are considered. In this chapter, we described the requirement to represent measurement on two levels; locally to the process, and at the global observation level. The reduction semantics does not consider this second level, and therefore we require *either* probabilistic branching *or* mixed states. But does it matter which?

In both cases, measurement results in a distribution of configurations; the probabilities or weights and the corresponding states are the same. The difference is in whether or not this distribution causes branching. The use of probabilistic branching in the reduction semantics is not necessarily wrong, as long as we are careful about the physical interpretation. In particular, it makes little sense to consider the observational properties of the quantum state in this setting. Conversion to mixed configurations is unlikely to offer any benefits related to interpretation, however it could be a worthwhile venture in order to maintain consistency between the two semantics.

Distributed modelling. CQP was developed by Gay and Nagarajan [2005] based on the idea that qubits are physical resources and processes represent distributed agents. One of the main results that arises from the type system, is the unique ownership of qubits. This result guarantees that each qubit is treated as a physical resource, and can only be in one place at any time.

It is worth considering whether this is a reasonable assumption, and what impact it has on observational equivalence. Using the distributed model does not allow us to consider concurrency in a physical locality. For example, a quantum computer may have concurrent processes, each with access to the same quantum registers. In this case, it may be required to model processes such as $\{q \ast\!= \mathsf{U}\}.P \parallel \{\mathsf{measure}\ q\}.Q$, in which parallel components share the same qubits.

Waiving the unique ownership of qubits condition would be incompatible with our notion of process equivalence. Included in the conditions for output matching, is the equality of the reduced density matrices of the qubits in the environment (ρ_E) after the output has occurred; this is the quantum subsystem directly available to an external process (the context). Instead, we would have to assume the context has access to the complete quantum state, thereby having more information available than is described by this reduced density matrix. This suggests that the concurrent approach used by Lalire [2006] does not agree with the use of the reduced density matrix in the process equivalence. This may partly explain why the bisimilarity in [Lalire 2006] is not preserved by parallel composition.

For the protocols that we are interested in, in particular communication protocols, it seems the logical approach to associate quantum systems with physically distributed locations. This enables us to consider, for example, what information an attacker can

gain with only a subset of the quantum system. Indeed, it should be possible to model local concurrency by implementing shared quantum registers as processes.

Language Extensions

The semantics presented in this chapter depends on each component of a mixed configuration having the same process structure. That means that only values may differ between components, and the result is that the complete configuration can make a *combined* transition. Although this is compatible with the current language, additions, such as the match operator, may not be.

Match operator. Incorporation of a match operator in CQP would provide the ability to model classical process control. The inclusion of conditional unitary operators is a partial solution that has enabled the formal modelling of protocols such as teleportation. Using a match operator, we might represent conditional operators as a guarded sum:

$$[x = 0]P + [x = 1]\{q *= \mathsf{X}\}.P$$

That is, if x evaluates to 0 continue as P, or if x evaluates to 1 then continue as $\{q *= \mathsf{X}\}.P$. Conditional operators represent only one type of classical control; for example, we could choose between totally different executions with a process $[x = 0]P + [x = 1]Q$.

The problem here, is that combined transitions become more complicated. In particular, not all components in a mixed configuration will have the same capabilities. Due to the possibility of producing observationally distinguishable executions, the match operator could be implemented as a second source of probabilistic branching. If, in the teleportation protocol, Bob were to use a match operator instead of a conditional unitary, we would get probabilistic branching in the execution. The difference between this, and the execution described in Chapter 3, is the point at which the branching occurs. Specifically, it would be after Bob has received the measurement values from Alice, and so this doesn't contradict quantum mechanics. Moreover, we would expect these branches to be observationally equivalent, and therefore we would still obtain the equivalence of teleportation and a quantum channel.

It may also be necessary to include the match operator in order to axiomatise bisimilarity in the presence of channel mobility. In particular, the match operator is used in the expansion law of the π-calculus in order to account for the substitution of channel names.

Recursion and replication. Another interesting extension to the language would be support for replication or recursion. This could provide, for example, the ability

to model a continuous EPR source. There are many protocols that can be modelled without using replication, or using finite replication for fixed length messages. However it is interesting to consider the impact of adding such features.

An example was given by Feng et al. [2006] which identifies one problem with recursive processes and quantum information; given a process $P = c![q].P$, the free qubits of P are not well-defined in qCCS. Indeed, such processes also result in typing contradictions in CQP. It is obvious that P is not a valid process, instead we are interested in processes such as $Q = c?[q].\{q *= \mathsf{H}\}.d![q].Q$, which do satisfy the typing rules.

The effect of recursion on process equivalence is of particular interest. Specifically, recursion may introduce loops into the transition system, although it is unclear what the effect of this would be. There has been a significant amount of recent work (for example, [Andova and Willemse 2006; Trčka and Georgievska 2008]) concerning the combination of probability and non-determinism in transition systems, and it is likely that this would be closely related to an implementation of recursion in CQP.

4.5.1 Comparison with qCCS

In this final section we discuss some of the differences between the language and equivalences presented in this chapter, and the work by Feng et al. [2011] on qCCS.

The language presented in [Feng et al. 2011] combines aspects from the models previously proposed in [Feng et al. 2007] and [Ying et al. 2009], and provides the ability to model processes with both quantum and classical data. A weak bisimilarity is defined for this new model of qCCS, which is found to be a congruence. Although this result addresses the same problem as we have been investigating in this chapter, there are several differences in the respective solutions.

The syntax of the two languages are considerably different; in qCSS there are fewer constructs and the syntax is defined inductively. The quantum operators in qCCS consist of superoperators ($\mathcal{E}[\widetilde{q}].P$) and measurements ($M[\widetilde{q}; x].P$); these exist as prefixes in qCCS as opposed to the action construct in CQP ($\{\cdot\}.P$) which may encompass quantum operators within an expression. Unitary operators and qubit initialisation are both cases of superoperators, however it is not possible for a qCCS process to initiate new qubits internally.

The ownership of qubits is controlled at the syntactic level in qCCS, through rules such as "If $qv(P) \cap qv(Q) = \emptyset$ then $P \parallel Q \in qProc$ and $qv(P \parallel Q) = qv(P) \cup qv(Q)$". This is in contrast to the type system in CQP which guarantees the unique ownership of qubits in typed processes, while not implementing any such restrictions at the syntactic level.

The type system is not only used to control qubit ownership, but also supports the flexible expression language of CQP. Although classical data can be represented

in qCCS, there is no corresponding language and semantics for the representation and evaluation of arbitrary expressions. This limits the ability to model protocols with classical components that require computation, however it can be argued that classical computations can be carried out using quantum data instead. As a result, the simpler language of qCCS could easily lead to more complicated process specifications, whereas CQP supports a higher-level description which is likely to be more appropriate for defining an accurate model in a formal verification setting.

A significant difference between the equivalences for CQP and qCCS is in the treatment of the quantum state. In qCCS, for two configurations to be bisimilar, the set of internal qubit names of the respective processes must be equal. Whereas in CQP the names, order and number of internal qubits can be different, and it is only upon output that the state is taken into account. Feng et al. [2011] also consider the verification of quantum teleportation, and this example illustrates the different treatments. In contrast to the direct quantum channel *QChannel* used in this chapter, the qCCS specification process features a 3-qubit unitary operator $SWAP_{1,3}$ which swaps the states of the first and third qubits. Their specification process is defined as

$$Tel_{spec} = c?[x].\{x, y, z *= SWAP_{1,3}\}.d![z].\mathbf{0}$$

expressed in the syntax of CQP. This process features three qubits instead of one, to match the three required by the teleportation process. The swap operator is also required in qCCS because the qubit names must be matched in the corresponding output actions, in contrast to our abstraction which is only concerned with the quantum state.

We argue that the additional qubits and the swap operation detract from the simplicity of the specification process, where effective verification using bisimilarity is dependent on an unambiguous specification. Although in the case of teleportation the behaviour of Tel_{spec} is obvious, this is unlikely to be the case for more complicated processes. For this reason, the bisimilarity of qCCS may be of limited use for verification.

Perhaps the most significant advantage of CQP is due to the introduction of mixed configurations in this chapter. Mixed configurations are similar to the configuration distributions used in qCCS as they both consist of distributions over pure configurations (i.e. process definition and quantum state). The difference with CQP is the incorporation of classical data in mixed configurations, which enables probabilistic branching to occur. As explained in this chapter, probabilistic branching may occur as the result of an output action in which the constituent components can be distinguished by the output values. Typically, this would arise because a measurement result influences the classical output, and hence the classical uncertainty of the state is reduced. A simple example is a quantum random number generator

$(\mathsf{qbit}\ x).\{x *= \mathsf{H}\}.c![\mathsf{measure}\ x].\mathbf{0}$, which outputs either 0 or 1. The output value will determine whether the quantum state is either $|0\rangle$ or $|1\rangle$ - it is no longer a mixed state and hence branching must be used.

In qCCS, all transitions are from (distributions of) configurations to distributions, and there is no concept of branching. In particular, it is not possible to have a transition in which the output values are not identical amongst the components of the distribution. As a result, the random number generator described above cannot be fully modelled in qCCS because the state after measurement, expressed in CQP as

$$\frac{1}{2} \bullet ([q \mapsto |0\rangle]; q; c![0].\mathbf{0}) \boxplus \frac{1}{2} \bullet ([q \mapsto |1\rangle]; q; c![1].\mathbf{0}) \ ,$$

would have no transitions. Ignoring such possibilities significantly reduces the ability of qCCS to represent the observational behaviour of processes. Furthermore, the qCCS bisimilarity could identify two processes incorrectly because a non-deterministic output is not acknowledged by the semantics. We therefore believe that qCCS is not able to accurately model the class of processes that contain such outputs, whereas the semantics of CQP have been designed specifically to capture this behaviour.

In comparison to qCCS, the proofs of preservation in this chapter are undoubtedly more involved, however a major cause of this is from the incorporation of mixed configurations with probabilistic branching in the semantics. As the example above illustrates, this added complexity is necessary to capture the full behaviour of quantum processes.

4.6 Summary

In Chapter 3, we found that the implementation of probabilistic branching results in observations of the quantum state that are incompatible with the theory of quantum mechanics. In light of these findings, we develop the semantics of CQP to produce a language in which the observational aspects of the quantum state agree with the mathematical formulation of quantum mechanics.

In Section 4.1, we introduce and motivate the use of *mixed configurations*, a concept that extends the notion of mixed quantum states to configurations. We use a series of examples to illustrate this novel approach combining internal branching and probabilistic branching. In Section 4.2, the new operational semantics are presented, including the transition relation for pure configurations (Figure 4.2), which is used to derive internal communication. The most significant change to the semantics is the source of probabilistic branching moving from R-Measure to L-Out. In particular, the rule L-Out now represents the probabilistic distribution of possible output values. Type preservation is reconsidered in Section 4.2.2 with respect to the new semantics.

Probabilistic branching bisimulation is redefined for the new semantics in Section 4.3. This relation is similar to the relation considered in Chapter 3, but takes into account the new behaviour arising from outputs. In Section 4.3.1, we prove that probabilistic branching bisimilarity is preserved by all contexts except input and qubit declaration prefixes. This includes preservation by parallel composition (Theorem 4.14), which has not been shown before for general quantum processes.

The step from bisimilarity to full bisimilarity, in order to achieve preservation by input prefix, is a method used in the π-calculus [Sangiorgi and Walker 2001]. We define *full probabilistic branching bisimilarity* in a similar fashion, by requiring bisimilarity to hold for all substitutions. In Section 4.3.2, we prove that full probabilistic branching bisimilarity is preserved by *all* contexts, and is therefore a congruence (Theorem 4.20).

In Section 4.4, we revisit quantum teleportation and show that it is *congruent* to a quantum channel based on the new semantics. We also show that the superdense coding protocol is congruent to a 2-bit classical channel.

5

Towards an Equational Theory

The notion of congruence is very important in process calculus because it provides the foundation for equational reasoning. An *axiomatisation* of an equivalence defines the rules that, together with the rules of equational reasoning (reflexivity, symmetry, and transitivity) prove *only* valid equations. An axiomatisation is *complete* if it proves *all* the valid equations.

In this chapter, we investigate the use of an axiomatic approach for proving process equivalence with respect to the full probabilistic branching bisimilarity defined in Chapter 4. Through the analysis of the teleportation protocol, specifically motivated by the equivalence $\mathit{Teleport} \leftrightarrows^c \mathit{QChannel}$ from Section 4.4, we identify a number of equalities with respect to the relation $\leftrightarrows^c$.

In contrast to axiomatisations for classical process calculi, we must consider the role of quantum operations. As a result, we present several rules for the manipulation of quantum operators, relating to familiar principles from quantum mechanics such as commutativity, deferred measurement, and implicit measurement. In Section 5.2 we prove the soundness of this axiomatisation.

Alongside rules for quantum operations we may expect to see rules for the structural manipulation of processes, analogous to the laws for classical process calculi. In Section 5.3, we consider the adaptation of the *expansion law* from the π-calculus, finding that the law cannot hold for CQP due to the semantics of expressions. As a result, the rules presented in this chapter do not form a complete axiomatisation.

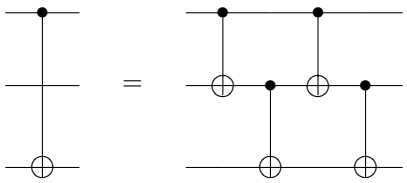


Figure 5.1. Controlled-NOT circuit identity.

5.1 Analysing Teleportation

The *expansion law* in process calculus enables the expansion of a parallel construction into a summation, in which each summand eliminates parallel composition at the top level. Repeated use of the law can *flatten* a process, the result being a summation of sequential processes where each summand corresponds to a single interleaving of parallel operations. As previously mentioned, a straightforward adaptation of the expansion law from the π-calculus is not possible, hence for the purpose of this analysis we begin with a flattened version of the teleportation protocol:

$$\begin{aligned} \mathit{Teleport} = (\mathsf{qbit}\ y, z).\{z \mathrel{*}\!= \mathsf{H}\}.\{z, y \mathrel{*}\!= \mathsf{CNot}\}.c?[x].\{x, z \mathrel{*}\!= \mathsf{CNot}\}. \\ \{x \mathrel{*}\!= \mathsf{H}\}.\{y \mathrel{*}\!= \mathsf{X}^{\mathsf{measure}\ z}\}.\{y \mathrel{*}\!= \mathsf{Z}^{\mathsf{measure}\ x}\}.d![y].\mathbf{0} \end{aligned} \quad (5.1)$$

In the following sections, we consider several analogues to circuit model identities, including commutativity, permutations, and implicit measurements. Through the application of these identities, we aim to simplify the quantum operators in the above teleportation process and transform it into the *QChannel* process.

5.1.1 Quantum Identities

In Section 3.3, we proved that teleportation is bisimilar to a state swapping circuit *Swap*. This result is a parallel to the analysis by Mermin [2001], in which equivalence is shown by using circuit model identities. These identities enable the gates to be moved around and replaced with other gates in the circuit. For example, Figure 5.1 illustrates how one controlled-NOT gate can be replaced by four, using an ancilla qubit. Another identity featured in [Mermin 2001] enables the control and target of a controlled-Z gate to be swapped; this is shown in Figure 5.2.

These identities are useful because they enable manipulation of the standard quantum gates, however this can be generalised to arbitrary quantum operators. If U, V, W are unitary operators, then we obtain the identity in Figure 5.3.

This circuit identity can be expressed by the following rule in CQP

$$\{\widetilde{x} \mathrel{*}\!= V\}.\{\widetilde{x} \mathrel{*}\!= W\}.P = \{\widetilde{x} \mathrel{*}\!= U\}.P \quad \text{if } U = WV\ . \qquad (\text{QI1})$$

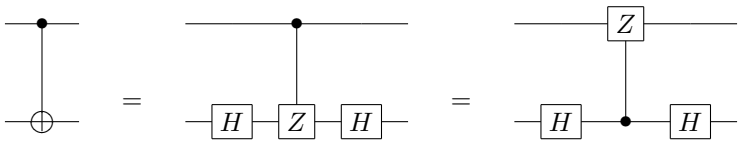


Figure 5.2. Controlled-Z circuit identity.

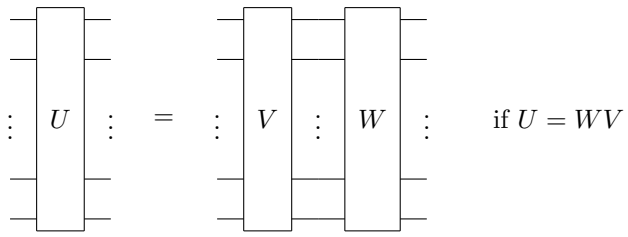


Figure 5.3. Circuit identity for arbitrary operators.

Applying QI1 to the teleportation process in (5.1) enables us to collapse the unitary actions into a single action. To combine single qubit gates with 2-qubit gates, we use the tensor product with the identity operator, for example, $\{z *= \mathsf{H}\} = \{z, y *= (\mathsf{H} \otimes \mathsf{I})\}$. This gives

$$(\mathsf{qbit}\ y, z).\{z, y *= \mathsf{CNot}.(\mathsf{H} \otimes \mathsf{I})\}.c?[x].\{x, z *= (\mathsf{H} \otimes \mathsf{I}).\mathsf{CNot}\}.$$
$$\{y *= \mathsf{X}^{\mathsf{measure}\ z}\}.\{y *= \mathsf{Z}^{\mathsf{measure}\ x}\}.d![y].\mathbf{0} \quad (5.2)$$

5.1.2 Deferred Measurement

The corrective operators X and Z are classically controlled by the outcomes of measuring qubits z and x. In Section 3.3, we showed an alternative implementation of the teleportation protocol ($Teleport_D$), in which these operators are replaced by quantum controlled operators. This is achieved through the *principle of deferred measurement* [Nielsen and Chuang 2000, p. 186]. We can express this principle as the following rule for an arbitrary unitary operator U:

$$\{\widetilde{y} *= U^{\mathsf{measure}\ x}\}.P = \{x, \widetilde{y} *= CU\}.\{\mathsf{measure}\ x\}.P \quad (\text{QI2})$$

where CU is a controlled-U gate.

If we apply QI2 to the two measurement operations in (5.2), noting that $\mathsf{CX} = \mathsf{CNot}$, we get

$$(\mathsf{qbit}\ y, z).\{z, y *= \mathsf{CNot}.(\mathsf{H} \otimes \mathsf{I})\}.c?[x].\{x, z *= (\mathsf{H} \otimes \mathsf{I}).\mathsf{CNot}\}.$$
$$\{z, y *= \mathsf{CNot}\}\{\mathsf{measure}\ z\}.\{x, y *= \mathsf{CZ}\}.\{\mathsf{measure}\ x\}.d![y].\mathbf{0}$$

5.1.3 Commuting Operators

In the circuit model, we are able to "slide" operators around due to commutativity. For example, we can swap the order of the measurement on z and the controlled-Z operator on x and y because the qubits are independent; mathematically, this is due to the use of the tensor product. For simplicity, we will only consider the basic quantum operations and not arbitrary expressions. The commutativity of internal operators are expressed by the following rules.

$$\{\widetilde{x} *= U\}.\{\widetilde{y} *= V\}.P = \{\widetilde{y} *= V\}.\{\widetilde{x} *= U\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC1}$$

$$\{\widetilde{x} *= U\}.\{\mathsf{measure}\ \widetilde{y}\}.P = \{\mathsf{measure}\ \widetilde{y}\}.\{\widetilde{x} *= U\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC2}$$

$$\{\widetilde{x} *= U\}.(\mathsf{qbit}\ \widetilde{y}).P = (\mathsf{qbit}\ \widetilde{y}).\{\widetilde{x} *= U\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC3}$$

$$\{\mathsf{measure}\ \widetilde{x}\}.\{\mathsf{measure}\ \widetilde{y}\}.P = \{\mathsf{measure}\ \widetilde{y}\}.\{\mathsf{measure}\ \widetilde{x}\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC4}$$

$$\{\mathsf{measure}\ \widetilde{x}\}.(\mathsf{qbit}\ \widetilde{y}).P = (\mathsf{qbit}\ \widetilde{y}).\{\mathsf{measure}\ \widetilde{x}\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC5}$$

$$(\mathsf{qbit}\ \widetilde{x}).(\mathsf{qbit}\ \widetilde{y}).P = (\mathsf{qbit}\ \widetilde{y}).(\mathsf{qbit}\ \widetilde{x}).P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC6}$$

Using QC2 on this, we can move the measurement of z after the controlled-Z operator, and then using QI1, the unitary operators are combined to give

$$(\mathsf{qbit}\ y, z).\{z, y *= \mathsf{CNot}.(\mathsf{H} \otimes \mathsf{I})\}.c?[x].\{x, y, z *= \mathsf{CZ}_{xy}.\mathsf{CNot}_{zy}.\mathsf{H}_x.\mathsf{CNot}_{xz}\}.$$
$$\{\mathsf{measure}\ z\}.\{\mathsf{measure}\ x\}.d![y].\mathbf{0}$$

where the subscripts on the unitary operators identify which qubits they apply to. The corresponding 3-qubit operators are found using a combination of permutations and tensor products with identity operators. For example, $\mathsf{CNot}_{xz} = \Pi^{-1}.(\mathsf{CNot} \otimes I).\Pi$, where π is a permutation given by $\pi(x) = x, \pi(y) = z, \pi(z) = y$, and Π is the permutation matrix corresponding to π.

The rules QC1–QC6 do not consider commutativity with input and output actions, which we would like to use in order to move the input action $c?[x]$ to the very top. In terms of the branching structure, the number of internal actions is not important, however we cannot simply swap an input and a unitary transformation as the following example demonstrates.

Example 5.1. Consider processes $\{x *= U\}.c?[y].\mathbf{0}$ and $c?[y].\{x *= U\}.\mathbf{0}$. If $\sigma = [q \mapsto |\psi\rangle]$ then, because x does not have to be included in every substitution, we have $(c?[y].\{x *= U\}.\mathbf{0}) \xrightarrow{c?[q]} \{x *= U\}.\mathbf{0})$, whilst $(\sigma; \emptyset; \{x *= U\}.c?[y].\mathbf{0})$ has no possible transitions. Therefore

$$\{x *= U\}.c?[y].\mathbf{0} \not\approx^c c?[y].\{x *= U\}.\mathbf{0}\ .$$

In the case of teleportation, the qubits y and z are initialised by the process so the situation in the example will never arise. Therefore we propose the rule

$$(\mathsf{qbit}\ \widetilde{x}).\{\widetilde{x} *= U\}.c?[\widetilde{y}].P \leftrightarrows^{c} (\mathsf{qbit}\ \widetilde{x}).c?[\widetilde{y}].\{\widetilde{x} *= U\}.P\ .$$

By applying this rule we can move the input action to the top of the process, giving

$$(\mathsf{qbit}\ y, z).c?[x].\{z, y *= \mathsf{CNot}.(\mathsf{H} \otimes \mathsf{I})\}.\{x, y, z *= \mathsf{CZ}_{xy}.\mathsf{CNot}_{zy}.\mathsf{H}_x.\mathsf{CNot}_{xz}\}.\\ \{\mathsf{measure}\ z\}.\{\mathsf{measure}\ x\}.d![y].\mathbf{0}\ .$$

We can generalise this rule to include several combinations of operations;

$$\alpha.\{\widetilde{y} *= U\}.c?[\widetilde{x}].P = \alpha.c?[\widetilde{x}].\{\widetilde{y} *= U\}.P \quad (\text{QC7})$$

$$\alpha.\{\widetilde{y} *= U\}.c![\widetilde{x}].P = \alpha.c![\widetilde{x}].\{\widetilde{y} *= U\}.P \quad (\text{QC8})$$

$$\alpha.\{\mathsf{measure}\ \widetilde{y}\}.c?[\widetilde{x}].P = \alpha.c?[\widetilde{x}].\{\mathsf{measure}\ \widetilde{y}\}.P \quad (\text{QC9})$$

$$\alpha.\{\mathsf{measure}\ \widetilde{y}\}.c![\widetilde{x}].P = \alpha.c![\widetilde{x}].\{\mathsf{measure}\ \widetilde{y}\}.P \quad (\text{QC10})$$

if $\widetilde{y} \subseteq \mathbf{n}(\alpha)$ and $\widetilde{y} \cap \widetilde{x} = \emptyset$. The condition $\widetilde{y} \in \mathbf{n}(\alpha)$ is necessary to ensure that the action on $\widetilde{y}$ does not result in blocking behaviour.

We are also able to commute qubit declarations with input and output actions since a qubit declaration is never blocking. This is expressed by the rules

$$(\mathsf{qbit}\ \widetilde{x}).c?[\widetilde{y}].P = c?[\widetilde{y}].(\mathsf{qbit}\ \widetilde{x}).P \text{ if } \widetilde{x} \cap \widetilde{y} = \emptyset \quad (\text{QC11})$$

$$(\mathsf{qbit}\ \widetilde{x}).c![\widetilde{y}].P = c![\widetilde{y}].(\mathsf{qbit}\ \widetilde{x}).P \text{ if } \widetilde{x} \cap \widetilde{y} = \emptyset\ . \quad (\text{QC12})$$

Using these rules we are able to bring the input action to the top, and also move the measurement operations after the output. This gives

$$c?[x].(\mathsf{qbit}\ y, z).\{z, y *= \mathsf{CNot}.(\mathsf{H} \otimes \mathsf{I})\}.\{x, y, z *= \mathsf{CZ}_{xy}.\mathsf{CNot}_{zy}.\mathsf{H}_x.\mathsf{CNot}_{xz}\}.\\ d![y].\{\mathsf{measure}\ z\}.\{\mathsf{measure}\ x\}.\mathbf{0}\ . \quad (5.3)$$

5.1.4 Surplus Operators

We have applied the principle of deferred measurement in order to swap classical control for quantum control. Now we consider the *principle of implicit measurement* ([Nielsen and Chuang 2000, p. 187]), which states that, any qubits at the end of a circuit may be assumed to be measured. We showed in Lemma 4.10 that a measurement does not affect the reduced density matrix of other qubits, hence we propose the rule

$$\{\mathsf{measure}\ x\}.\mathbf{0} = \mathbf{0}\ . \quad (\text{QS1})$$

Then, by QS1, we can eliminate the measurements in (5.3). By combining the remaining quantum operators with QI1, we obtain

$$c?[x].(\mathsf{qbit}\ y, z).\{x, y, z *= \mathsf{CZ}_{xy}.\mathsf{CNot}_{zy}.\mathsf{H}_x.\mathsf{CNot}_{xz}.\mathsf{CNot}_{zy}.\mathsf{H}_z\}.d![y].\mathbf{0}\ . \tag{5.4}$$

Measurements are not the only operators that produce no observable effect at the end of a process – the same applies to unitary operators and qubit declarations, as expressed by the rules

$$\{\widetilde{x} *= U\}.\mathbf{0} = \mathbf{0} \tag{QS2}$$

$$(\mathsf{qbit}\ x).\mathbf{0} = \mathbf{0}\ . \tag{QS3}$$

5.1.5 Permutations

Up until this point, we have applied rules that, in general, simplify the process. Proving that (5.4) is bisimilar to *QChannel* is considerably easier than for the original teleportation process, because many intermediate states have been eliminated. However, it is not obvious how this can be shown equationally. In particular, the ancilla qubits that are used in teleportation must be eliminated, since they are not present in *QChannel*.

First, we shall introduce extra unitary operators at the end. We can see that qubits x and z will each finish in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and so we apply the Hadamard operator to each. The rule QS2 given in the previous section allows these operations to be added, thus giving

$$c?[x].(\mathsf{qbit}\ y, z).\{x, y, z *= \mathsf{CZ}_{xy}.\mathsf{CNot}_{zy}.\mathsf{H}_x.\mathsf{CNot}_{xz}.\mathsf{CNot}_{zy}.\mathsf{H}_z\}.$$
$$d![y].\{x *= \mathsf{H}\}.\{z *= \mathsf{H}\}.\mathbf{0}\ .$$

The operators can then be combined into a single unitary action by using QC8 and QI1;

$$c?[x].(\mathsf{qbit}\ y, z).\{x, y, z *= \mathsf{H}_x.\mathsf{H}_z.\mathsf{CZ}_{xy}.\mathsf{CNot}_{zy}.\mathsf{H}_x.\mathsf{CNot}_{xz}.\mathsf{CNot}_{zy}.\mathsf{H}_z\}.d![y].\mathbf{0}\ . \tag{5.5}$$

Next, we insert a permutation in order to swap the output qubit y with x. Let π be a permutation of qubits, and let Π be the corresponding permutation on the quantum state, then define the rule

$$\alpha.P = \alpha.\{\widetilde{x} *= \Pi\}.P\{\pi(\widetilde{q})/\widetilde{x}\} \text{ if } \widetilde{x} \subseteq \mathbf{n}(\alpha)\ . \tag{QP1}$$

As in previous rules, we require $\widetilde{x} \subseteq \mathbf{n}(\alpha)$ in order to prevent the introduction of a

blocking operator. Applying this rule to (5.5), followed by QI1, we get

$$c?[x].(\mathsf{qbit}\ y,z).\{x,y,z \mathrel{*}\!= \Pi.\mathsf{H}_x.\mathsf{H}_z.\mathsf{CZ}_{xy}.\mathsf{CNot}_{zy}.\mathsf{H}_x.\mathsf{CNot}_{xz}.\mathsf{CNot}_{zy}.\mathsf{H}_z\}.d![x].\mathbf{0}\ . \tag{5.6}$$

where $\pi(x) = y, \pi(y) = x, \pi(z) = z$ and Π is the corresponding permutation operator.

5.1.6 Qubit Declaration

In this final stage, we will simplify the unitary operation by taking account of the qubit declaration $(\mathsf{qbit}\ y, z)$. This declaration ensures that the quantum state of the new qubits y, z will be $|00\rangle$, and hence the input domain of the unitary operation is significantly restricted.

Let

$$U = \Pi.\mathsf{H}_x.\mathsf{H}_z.\mathsf{CZ}_{xy}.\mathsf{CNot}_{zy}.\mathsf{H}_x.\mathsf{CNot}_{xz}.\mathsf{CNot}_{zy}.\mathsf{H}_z\ .$$

Then we have

$$c?[x].(\mathsf{qbit}\ y,z).\{x,y,z \mathrel{*}\!= U\}.d![x].\mathbf{0}\ .$$

In this process, we have the qubit declaration $(\mathsf{qbit}\ yz)$ which introduces two qubits in the combined state $|00\rangle$. We can define a linear map Q corresponding to this declaration:

$$Q = I \otimes \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1&0\\0&0\\0&0\\0&0\\0&1\\0&0\\0&0\\0&0\end{bmatrix}\ .$$

Then, the action of teleportation on the single qubit x is given by UQ. Based on QI1, we can define a similar rule to deal with quantum operators that appear under qubit declarations.

$$(\mathsf{qbit}\ x).\{\widetilde{y}x \mathrel{*}\!= U\}.P \leftrightarrows^c (\mathsf{qbit}\ x).\{\widetilde{y}x \mathrel{*}\!= V\}.P \text{ if } U(I_{\widetilde{y}} \otimes |0\rangle) = V(I_{\widetilde{y}} \otimes |0\rangle)\ . \quad (\text{QD1})$$

We have $UQ = \mathsf{I}_{xyz}Q$ where I_{xyz} is the identity operator on qubits x, y, z. Therefore, by applying QD1 to (5.6), we have

$$c?[x].(\mathsf{qbit}\ y,z).\{x,y,z \mathrel{*}\!= \mathsf{I}\}.d![x].\mathbf{0}\ . \tag{5.7}$$

From this point, we can apply QI1, QC8, and QS2 to give

$$c?[x].\{x *= \mathsf{I}\}.d![x].\mathbf{0} \ .$$

Finally, a special case of QP1, in which we consider the identity permutation, results in the process that we are aiming for:

$$c?[x].d![x].\mathbf{0} \ .$$

5.2 Soundness of the Equational Laws

In the previous section we presented a number of equalities with respect to full probabilistic branching bisimulation; these are summarised in Figure 5.4. In this section, we prove the soundness of these laws. Included with these rules are the laws R1–R3 for manipulating restrictions. These laws were not used in the analysis of teleportation in the previous section, however they are common laws for classical process calculi, and their use will be demonstrated in Section 5.3.

Each proof in this section follows the same argument; we define a suitable equivalence relation for the processes in question, and then prove that it is a probabilistic branching bisimulation for arbitrary quantum states and substitutions.

Quantum identities

This lemma proves the soundness of QI1.

Lemma 5.1 (Operator identities)**.** *For any process P and unitary operators U, W, V, if $U = WV$ then*

$$\{\widetilde{y} *= V\}.\{\widetilde{y} *= W\}.P \leftrightarrows^c \{\widetilde{y} *= U\}.P \ .$$

Proof. Let $\kappa = \{\widetilde{u}, \widetilde{p}, \widetilde{q}/\widetilde{x}, \widetilde{y}, \widetilde{z}\}$ be an arbitrary substitution; we assume that the variables $\widetilde{y}$ are included, otherwise there are no possible transitions. Let $\sigma_1 = [\widetilde{pr} \mapsto |\psi_1\rangle]$, $\sigma_2 = [\widetilde{pr} \mapsto |\psi_2\rangle]$, and $\sigma_3 = [\widetilde{pr} \mapsto |\psi_3\rangle]$, where $|\psi_2\rangle = (V \otimes I_{\widetilde{r}})|\psi_1\rangle$ and $|\psi_3\rangle = (W \otimes I_{\widetilde{r}})|\psi_2\rangle$. Because $U = WV$ we have $|\psi_3\rangle = (U \otimes I_{\widetilde{r}})|\psi_1\rangle$. Now let

$$\begin{aligned}
s_1 &= (\sigma_1; \widetilde{p}, \widetilde{q}; (\{\widetilde{p} *= U\}.P)\kappa), \\
s_2 &= (\sigma_1; \widetilde{p}, \widetilde{q}; (\{\widetilde{p} *= V\}.\{\widetilde{p} *= W\}.P)\kappa), \\
s_3 &= (\sigma_2; \widetilde{p}, \widetilde{q}; (\{\widetilde{p} *= W\}.P)\kappa), \\
s_4 &= (\sigma_3; \widetilde{p}, \widetilde{q}; P\kappa) \ .
\end{aligned}$$

Define an equivalence relation $\mathcal{R}$ as

$$\mathcal{R} = \{(s_1, s_2), (s_1, s_3), (s_2, s_3)\} \cup \mathcal{I}$$

$$\{\widetilde{x} *= V\}.\{\widetilde{x} *= W\}.P = \{\widetilde{x} *= U\}.P \qquad \text{if } U = WV \tag{QI1}$$

$$\{\widetilde{y} *= \mathsf{U}^{\mathsf{measure}\ x}\}.P = \{x, \widetilde{y} *= \mathsf{CU}\}.\{\mathsf{measure}\ x\}.P \tag{QI2}$$

$$\{\widetilde{x} *= U\}.\{\widetilde{y} *= V\}.P = \{\widetilde{y} *= V\}.\{\widetilde{x} *= U\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC1}$$

$$\{\widetilde{x} *= U\}.\{\mathsf{measure}\ \widetilde{y}\}.P = \{\mathsf{measure}\ \widetilde{y}\}.\{\widetilde{x} *= U\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC2}$$

$$\{\widetilde{x} *= U\}.(\mathsf{qbit}\ \widetilde{y}).P = (\mathsf{qbit}\ \widetilde{y}).\{\widetilde{x} *= U\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC3}$$

$$\{\mathsf{measure}\ \widetilde{x}\}.\{\mathsf{measure}\ \widetilde{y}\}.P = \{\mathsf{measure}\ \widetilde{y}\}.\{\mathsf{measure}\ \widetilde{x}\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC4}$$

$$\{\mathsf{measure}\ \widetilde{x}\}.(\mathsf{qbit}\ \widetilde{y}).P = (\mathsf{qbit}\ \widetilde{y}).\{\mathsf{measure}\ \widetilde{x}\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC5}$$

$$(\mathsf{qbit}\ \widetilde{x}).(\mathsf{qbit}\ \widetilde{y}).P = (\mathsf{qbit}\ \widetilde{y}).(\mathsf{qbit}\ \widetilde{x}).P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC6}$$

$$\alpha.\{\widetilde{y} *= U\}.c?[\widetilde{x}].P = \alpha.c?[\widetilde{x}].\{\widetilde{y} *= U\}.P \quad \text{if } \widetilde{y} \subseteq \mathbf{n}(\alpha), \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC7}$$

$$\alpha.\{\widetilde{y} *= U\}.c![\widetilde{x}].P = \alpha.c![\widetilde{x}].\{\widetilde{y} *= U\}.P \quad \text{if } \widetilde{y} \subseteq \mathbf{n}(\alpha), \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC8}$$

$$\alpha.\{\mathsf{measure}\ \widetilde{y}\}.c?[\widetilde{x}].P = \alpha.c?[\widetilde{x}].\{\mathsf{measure}\ \widetilde{y}\}.P \quad \text{if } \widetilde{y} \subseteq \mathbf{n}(\alpha), \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC9}$$

$$\alpha.\{\mathsf{measure}\ \widetilde{y}\}.c![\widetilde{x}].P = \alpha.c![\widetilde{x}].\{\mathsf{measure}\ \widetilde{y}\}.P \quad \text{if } \widetilde{y} \subseteq \mathbf{n}(\alpha), \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC10}$$

$$(\mathsf{qbit}\ \widetilde{x}).c?[\widetilde{y}].P = c?[\widetilde{y}].(\mathsf{qbit}\ \widetilde{x}).P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC11}$$

$$(\mathsf{qbit}\ \widetilde{x}).c![\widetilde{y}].P = c![\widetilde{y}].(\mathsf{qbit}\ \widetilde{x}).P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC12}$$

$$\{\mathsf{measure}\ x\}.\mathbf{0} = \mathbf{0} \tag{QS1}$$

$$\{x *= U\}.\mathbf{0} = \mathbf{0} \tag{QS2}$$

$$(\mathsf{qbit}\ x).\mathbf{0} = \mathbf{0} \tag{QS3}$$

$$\alpha.\{\widetilde{x} *= \Pi\}.P\{\pi(\widetilde{q})/\widetilde{x}\} = \alpha.P \quad \text{if } \widetilde{x} \subseteq \mathbf{n}(\alpha) \tag{QP1}$$

$$(\mathsf{qbit}\ x).\{\widetilde{y}x *= U\}.P = (\mathsf{qbit}\ x).\{\widetilde{y}x *= V\}.P \quad \text{if } U(I_{\widetilde{y}} \otimes |0\rangle) = V(I_{\widetilde{y}} \otimes |0\rangle) \tag{QD1}$$

$$(\nu c)(P + Q) = (\nu c)P + (\nu c)Q \tag{R1}$$

$$(\nu c)\alpha.P = \mathbf{0} \qquad \text{if } \alpha \in \{c?[\cdot], c![\cdot]\} \tag{R2}$$

$$(\nu c)\alpha.P = \alpha.(\nu c)P \qquad \text{if } \alpha \notin \{c?[\cdot], c![\cdot]\} \tag{R3}$$

Figure 5.4. Axioms for full probabilistic branching bisimilarity.

where $\mathcal{I}$ is the identity relation. We now prove that $\mathcal{R}$ is a probabilistic branching bisimulation by case analysis of the transitions of s_1, s_2, and s_3.

If $s_1 \xrightarrow{\tau} s_4$ then we have $s_2 \xrightarrow{\tau}\xrightarrow{\tau} s_4$ and also $s_3 \xrightarrow{\tau} s_4$ where $(s_4, s_4) \in \mathcal{R}$. If $s_2 \xrightarrow{\tau} s_3$ then we have $(s_1, s_3) \in \mathcal{R}$ and $(s_3, s_3) \in \mathcal{R}$. If $s_3 \xrightarrow{\tau} s_4$ then we have $s_1 \xrightarrow{\tau} s_4$ and also $s_2 \xrightarrow{\tau}\xrightarrow{\tau} s_4$ where $(s_4, s_4) \in \mathcal{R}$. □

Deferred measurement

This lemma proves the soundness of QI2.

Lemma 5.2 (Deferred measurement)**.** *Assume $x \notin \widetilde{y}$. If U is a unitary operator and CU is the corresponding controlled operator then*

$$\{\widetilde{y} *= U^{\mathsf{measure}\ x}\}.P \leftrightarrows^c \{x, \widetilde{y} *= CU\}.\{\mathsf{measure}\ x\}.P\ .$$

Proof. Assume that $\kappa = \{p, \widetilde{q}/x, \widetilde{y}\}$. The substitution of more qubits has no effect on the proof, whilst fewer qubits will result in some configurations blocking; in the latter case a simpler relation is required. Let

$$\begin{aligned}
s_1 &= ([p\widetilde{q} \mapsto |\psi_1\rangle]; p, \widetilde{q}; (\{\widetilde{y} *= U^{\mathsf{measure}\ x}\}.P)\kappa),\\
s_2 &= \oplus_{i\in I}\ g_i\ ([p\widetilde{q} \mapsto |\psi_{2_i}\rangle]; p, \widetilde{q}; \lambda z.(\{\widetilde{y} *= U^z\}.P)\kappa; i),\\
s_3 &= \oplus_{i\in I}\ g_i\ ([p\widetilde{q} \mapsto |\psi_{3_i}\rangle]; p, \widetilde{q}; P\kappa),\\
s_4 &= ([p\widetilde{q} \mapsto |\psi_1\rangle]; p, \widetilde{q}; (\{x, \widetilde{y} *= CU\}.\{\mathsf{measure}\ x\}.P)\kappa),\\
s_5 &= ([p\widetilde{q} \mapsto |\psi_4\rangle]; p, \widetilde{q}; (\{\mathsf{measure}\ x\}.P)\kappa),\\
s_6 &= \oplus_{i\in I}\ g_i\ ([p\widetilde{q} \mapsto |\psi_{5_i}\rangle]; p, \widetilde{q}; P\kappa)
\end{aligned}$$

where $I = \{0, 1\}$. Let M_0, M_1 be the measurement operators corresponding to the measurement of x, then

$$M_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \otimes I_{\widetilde{y}} \text{ and } M_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes I_{\widetilde{y}}\ .$$

The controlled operator is defined by the matrix

$$CU = \begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix}\ .$$

Then $|\psi_{2_i}\rangle = M_i|\psi_1\rangle$ and $|\psi_{3_i}\rangle = U^i|\psi_{2_i}\rangle = U^iM_i|\psi_1\rangle$ and $|\psi_4\rangle = CU|\psi_1\rangle$ and $|\psi_{5_i}\rangle = M_i|\psi_4\rangle = M_iCU|\psi_1\rangle$. For each $i \in I$ a straightforward calculation shows $U^iM_i = M_iCU$, therefore $|\psi_{3_i}\rangle = |\psi_{5_i}\rangle$ and $s_3 = s_6$.

Now define an equivalence relation where

$$\mathcal{R} = \{(s_1, s_2), (s_1, s_4), (s_1, s_5)\} \cup \mathcal{I} \ .$$

We have $s_1 \xrightarrow{\tau} s_2 \xrightarrow{\tau} s_3$ and $s_4 \xrightarrow{\tau} s_5 \xrightarrow{\tau} s_6$. Therefore it is straightforward to see that $\mathcal{R}$ is a probabilistic branching bisimulation. □

Commuting operators

In this lemma we prove the soundness of the rules QC1–QC6, relating to the commutativity of internal actions.

Lemma 5.3 (Internal commutativity)**.** *If* $\widetilde{x} \cap \widetilde{y} = \emptyset$ *then*

1. $\{\widetilde{x} *= U\}.\{\widetilde{y} *= V\}.P \leftrightarrows^c \{\widetilde{y} *= V\}.\{\widetilde{x} *= U\}.P.$
2. $\{\widetilde{x} *= U\}.\{\mathsf{measure}\ \widetilde{y}\}.P \leftrightarrows^c \{\mathsf{measure}\ \widetilde{y}\}.\{\widetilde{x} *= U\}.P.$
3. $\{\widetilde{x} *= U\}.(\mathsf{qbit}\ \widetilde{y}).P \leftrightarrows^c (\mathsf{qbit}\ \widetilde{y}).\{\widetilde{x} *= U\}.P.$
4. $\{\mathsf{measure}\ \widetilde{x}\}.\{\mathsf{measure}\ \widetilde{y}\}.P \leftrightarrows^c \{\mathsf{measure}\ \widetilde{y}\}.\{\mathsf{measure}\ \widetilde{x}\}.P.$
5. $\{\mathsf{measure}\ \widetilde{x}\}.(\mathsf{qbit}\ \widetilde{y}).P \leftrightarrows^c (\mathsf{qbit}\ \widetilde{y}).\{\mathsf{measure}\ \widetilde{x}\}.P.$
6. $(\mathsf{qbit}\ \widetilde{x}).(\mathsf{qbit}\ \widetilde{y}).P \leftrightarrows^c (\mathsf{qbit}\ \widetilde{y}).(\mathsf{qbit}\ \widetilde{x}).P.$

Proof. For each case we construct an equivalence relation and then prove it is a bisimulation. We consider the case $\{\widetilde{x}*=U\}.\{\mathsf{measure}\ \widetilde{y}\}.P \leftrightarrows^c \{\mathsf{measure}\ \widetilde{y}\}.\{\widetilde{x}*=U\}.P$ and note that the other cases are similar.

Let

$$\begin{aligned}
s_1 &= ([\widetilde{q} \mapsto |\psi_1\rangle]; \omega_1; (\{\widetilde{x} *= U\}.\{\mathsf{measure}\ \widetilde{y}\}.P)\kappa),\\
s_2 &= ([\widetilde{q} \mapsto |\psi_2\rangle]; \omega_1; (\{\mathsf{measure}\ \widetilde{y}\}.P)\kappa),\\
s_3 &= ([\widetilde{q} \mapsto |\psi_1\rangle]; \omega_1; (\{\mathsf{measure}\ \widetilde{y}\}.\{\widetilde{x} *= U\}.P)\kappa),\\
s_4 &= \oplus_{i\in I}\ g_i\ ([\widetilde{q} \mapsto |\psi_{3_i}\rangle]; \omega_1; (\{\widetilde{x} *= U\}.P)\kappa),\\
s_5 &= \oplus_{i\in I}\ g_i\ ([\widetilde{q} \mapsto |\psi_{4_i}\rangle]; \omega_1; P\kappa)\ .
\end{aligned}$$

Assume that $\widetilde{x}$ and $\widetilde{y}$ are included in κ (otherwise a different relation is required since not all configurations admit a transition). Let U_x denote U applied to qubits x and let $\{M_i\}$ be the measurement operators for qubits $\widetilde{y}$. We have $s_1 \xrightarrow{\tau}\xrightarrow{\tau} s_2 \xrightarrow{\tau}\xrightarrow{\tau} s_5$ where $|\psi_2\rangle = U_x|\psi_1\rangle$ and $|\psi_{4_i}\rangle = M_i|\psi_2\rangle$. Because U_x and M_i act on different qubits, we have $U_xM_i = M_iU_x$. Therefore $s_3 \xrightarrow{\tau}\xrightarrow{\tau} s_4 \xrightarrow{\tau}\xrightarrow{\tau} s_5$ where $|\psi_{3_i}\rangle = M_i|\psi_1\rangle$ and $|\psi_{4_i}\rangle = U_x|\psi_{3_i}\rangle$. Define an equivalence relation $\mathcal{R}$ by taking the equivalence closure of

$$\mathcal{R} = \{(s_1, s_2), (s_1, s_3), (s_1, s_4)\} \cup \mathcal{I}$$

where $\mathcal{I}$ is the identity relation. It is straightforward to see that $\mathcal{R}$ is a probabilistic branching bisimulation. □

The rules QC7–QC12 must be treated differently than the other commutative laws due to the presence of visible actions. In these cases, the inclusion of the prefix α plays a central role in the proof, ensuring that the quantum operation is not blocking.

Lemma 5.4 (External action commutativity)**.** *If $\widetilde{y} \subseteq \mathbf{n}(\alpha)$ and $\widetilde{x} \cap \widetilde{y} = \emptyset$ then*

1. $\alpha.\{\widetilde{y} *= U\}.c?[\widetilde{x}].P \leftrightarrows^c \alpha.c?[\widetilde{x}].\{\widetilde{y} *= U\}.P.$
2. $\alpha.\{\widetilde{y} *= U\}.c![\widetilde{x}].P \leftrightarrows^c \alpha.c![\widetilde{x}].\{\widetilde{y} *= U\}.P.$
3. $\alpha.\{\mathsf{measure}\ \widetilde{y}\}.c?[\widetilde{x}].P \leftrightarrows^c \alpha.c?[\widetilde{x}].\{\mathsf{measure}\ \widetilde{y}\}.P.$
4. $\alpha.\{\mathsf{measure}\ \widetilde{y}\}.c![\widetilde{x}].P \leftrightarrows^c \alpha.c![\widetilde{x}].\{\mathsf{measure}\ \widetilde{y}\}.P.$
5. $(\mathsf{qbit}\ \widetilde{x}).c?[\widetilde{y}].P \leftrightarrows^c c?[\widetilde{y}].(\mathsf{qbit}\ \widetilde{x}).P.$
6. $(\mathsf{qbit}\ \widetilde{x}).c![\widetilde{y}].P \leftrightarrows^c c![\widetilde{y}].(\mathsf{qbit}\ \widetilde{x}).P.$

Proof. In cases 1–4, the condition $\widetilde{y} \subseteq \mathbf{n}(\alpha)$ ensures that the expression evaluation is not blocking if the term α is not blocking. In cases 5 and 6, this condition is not required because the qubit declaration is never blocking. We consider the case $\alpha.\{\widetilde{y} *= U\}.c![\widetilde{x}].P \leftrightarrows^c \alpha.c![\widetilde{x}].\{\widetilde{y} *= U\}.P$ and note that the other cases are similar.

Let κ be an arbitrary substitution, σ_1 be an arbitrary quantum state, and let

$$
\begin{aligned}
s_1 &= (\sigma_1; \omega_1; (\alpha.\{\widetilde{y} *= U\}.c![\widetilde{x}].P)\kappa_1) \\
s_2 &= (\sigma_2; \omega_2; (\{\widetilde{y} *= U\}.c![\widetilde{x}].P)\kappa_1\kappa_2) \\
s_3 &= (\sigma_3; \omega_2; (c![\widetilde{x}].P)\kappa_1\kappa_2) \\
s_4 &= (\sigma_3; \omega_3; P\kappa_1\kappa_2) \\
t_1 &= (\sigma_1; \omega_1; (\alpha.c![\widetilde{x}].\{\widetilde{y} *= U\}.P)\kappa_1) \\
t_2 &= (\sigma_2; \omega_2; (c![\widetilde{x}].\{\widetilde{y} *= U\}.P)\kappa_1\kappa_2) \\
t_3 &= (\sigma_3; \omega_3; (\{\widetilde{y} *= U\}.P)\kappa_1\kappa_2) \\
t_4 &= (\sigma_3; \omega_3; P\kappa_1\kappa_2)
\end{aligned}
$$

where κ_2 is a substitution due to α. Assume that $\kappa_1\kappa_2 = \{\widetilde{p}, \widetilde{q}/\widetilde{x}, \widetilde{y}\}$, then we have the transitions $s_1 \xrightarrow{\alpha} s_2 \xrightarrow{\tau} s_3 \xrightarrow{c![\widetilde{p}]} s_4$ and $t_1 \xrightarrow{\alpha} t_2 \xrightarrow{c![\widetilde{p}]} t_3 \xrightarrow{\tau} t_4$. We note that $s_4 = t_4$, and that $\rho_E(s_4) = \rho_E(t_3)$ because unitary operators are trace-preserving. Therefore, we define an equivalence relation

$$\mathcal{R} = \{(s_1, t_1), (s_2, t_2), (s_3, t_3), (s_4, t_3)\} \cup \mathcal{I}$$

where $\mathcal{I}$ is the identity relation. Then it is straightforward to see that $\mathcal{R}$ is a probabilistic branching bisimulation. □

Surplus operators

The soundness of the rules relating to surplus operators, Qs1–Qs3, are proved in the following lemma.

Lemma 5.5. *1.* $\{\mathsf{measure}\ \widetilde{x}\}.\mathbf{0} \leftrightarrows^c \mathbf{0}$.

2. $\{\widetilde{x} *= U\}.\mathbf{0} \leftrightarrows^c \mathbf{0}$.

3. $(\mathsf{qbit}\ x).\mathbf{0} \leftrightarrows^c \mathbf{0}$.

Proof. For each case we construct a relation and prove that it is a probabilistic branching bisimulation.

1. Let κ be an arbitrary substitution, σ_1 be an arbitrary quantum state, and let

$$s_1 = (\sigma_1; \omega; (\{\mathsf{measure}\ \widetilde{x}\}.\mathbf{0})\kappa),\ s_2 = \oplus_{i\in I}\ g_i\ (\sigma_{2_i}; \omega; \mathbf{0}),\ \text{and}\ s_3 = (\sigma_1; \omega; \mathbf{0})\ .$$

   Then define an equivalence relation $\mathcal{R}$ where

$$\mathcal{R} = \{(s_1, s_2), (s_1, s_3), (s_2, s_3)\}\ .$$

   If qubits $\widetilde{x}$ are replaced by κ then we have the transition $s_1 \xrightarrow{\tau} s_2$ and $(s_2, s_3) \in \mathcal{R}$. There are no transitions in all other cases.

2. For arbitrary σ_1 and κ, let

$$s_1 = (\sigma_1; \omega; (\{\widetilde{x} *= U\}.\mathbf{0})\kappa),\ s_2 = (\sigma_2; \omega; \mathbf{0}),\ \text{and}\ s_3 = (\sigma_1; \omega; \mathbf{0})\ .$$

   Then define an equivalence relation $\mathcal{R}$ where

$$\mathcal{R} = \{(s_1, s_2), (s_1, s_3), (s_2, s_3)\}\ .$$

   If qubits $\widetilde{x}$ are replaced by κ then we have the transition $s_1 \xrightarrow{\tau} s_2$ and $(s_2, s_3) \in \mathcal{R}$. There are no transitions in all other cases.

3. For arbitrary σ_1 and κ, let

$$s_1 = (\sigma_1; \omega; ((\mathsf{qbit}\ x).\mathbf{0})\kappa),\ s_2 = (\sigma_2; \omega; \mathbf{0}),\ \text{and}\ s_3 = (\sigma_1; \omega; \mathbf{0})\ .$$

   Then define an equivalence relation $\mathcal{R}$ where

$$\mathcal{R} = \{(s_1, s_2), (s_1, s_3), (s_2, s_3)\}\ .$$

The variable x is bound in s_1, hence κ has no effect. We have the transition $s_1 \xrightarrow{\tau} s_2$ and $(s_2, s_3) \in \mathcal{R}$. The configurations s_2 and s_3 admit no transitions.

□

Permutations

This next lemma proves the soundness of QP1.

Lemma 5.6 (Permutations)**.** *If π is a permutation on qubit variables $\widetilde{x}$ and Π is the corresponding permutation operator and $\widetilde{x} \subseteq \mathbf{n}(\alpha)$ then*

$$\alpha.\{\widetilde{x} *= \Pi\}.P\{\pi(\widetilde{x})/\widetilde{x}\} \leftrightarrows^c \alpha.P \ .$$

Proof. Assume that $\kappa_1 = \{\widetilde{q}/\widetilde{x}\}$; if not all qubits $\widetilde{x}$ are included then neither process can proceed, on the other hand substituting more qubits has no effect on the proof. Let $\sigma_1 = [\widetilde{q} \mapsto |\psi\rangle]$, and let $\sigma_2 = [\pi(\widetilde{q}) \mapsto \Pi|\psi\rangle]$. Now let

$$\begin{aligned}
s_1 &= (\sigma_1; \widetilde{q}; (\alpha.\{\widetilde{x} *= \Pi\}.P\{\pi(\widetilde{x})/\widetilde{x}\})\kappa_1), \\
s_2 &= (\sigma_2; \widetilde{q}'; (\{\widetilde{x} *= \Pi\}.P\{\pi(\widetilde{x})/\widetilde{x}\})\kappa_1\kappa_2), \\
s_3 &= (\sigma_3; \widetilde{q}'; P\{\pi(\widetilde{x})/\widetilde{x}\}\kappa_1\kappa_2), \\
t_1 &= (\sigma_1; \widetilde{q}; \alpha.P\kappa_1), \\
t_2 &= (\sigma_2; \widetilde{q}'; P\kappa_1\kappa_2)
\end{aligned}$$

where κ_2 is a substitution introduced by α. Then we have the transitions $s_1 \xrightarrow{\alpha} s_2 \xrightarrow{\tau} s_3$ and $t_1 \xrightarrow{\alpha} t_2$. Define an equivalence relation as

$$\mathcal{R} = \{(s_1, t_1), (s_2, s_3)\} \cup \mathcal{I}$$

where $\mathcal{I}$ is the identity relation. Structural congruence of configurations gives $s_3 \equiv t_2$, therefore $(s_3, t_2) \in \mathcal{R}$. It is then straightforward to see that $\mathcal{R}$ is a probabilistic branching bisimulation. □

Qubit declaration

The soundness of QD1 is proved in the following lemma.

Lemma 5.7 (Qubit declaration)**.** *If $U(I_{\widetilde{y}} \otimes |0\rangle) = V(I_{\widetilde{y}} \otimes |0\rangle)$ then*

$$(\mathsf{qbit}\ x).\{\widetilde{y}x *= U\}.P \leftrightarrows^c (\mathsf{qbit}\ x).\{\widetilde{y}x *= V\}.P \ .$$

Proof. Assume that $\kappa = \{\widetilde{q}/\widetilde{y}\}$. Let $\sigma_1 = [\widetilde{q} \mapsto |\psi\rangle]$, $\sigma_2 = [\widetilde{q}r \mapsto |\psi\rangle|0\rangle]$, and

$\sigma_3 = [\widetilde{q}r \mapsto U|\psi\rangle|0\rangle]$. We have $|\psi\rangle|0\rangle = (I_{\widetilde{y}} \otimes |0\rangle)|\psi\rangle$, hence $U|\psi\rangle|0\rangle = V|\psi\rangle|0\rangle$. Let

$$\begin{aligned}
s_1 &= (\sigma_1; \widetilde{q}; ((\mathsf{qbit}\ x).\{\widetilde{y}x \mathrel{*}= U\}.P)\kappa),\\
s_2 &= (\sigma_2; \widetilde{q}, r; (\{\widetilde{y}r \mathrel{*}= U\}.P\{r/x\})\kappa),\\
s_3 &= (\sigma_1; \widetilde{q}; ((\mathsf{qbit}\ x).\{\widetilde{y}x \mathrel{*}= V\}.P)\kappa),\\
s_4 &= (\sigma_2; \widetilde{q}, r; (\{\widetilde{y}r \mathrel{*}= V\}.P\{r/x\})\kappa),\\
s_5 &= (\sigma_3; \widetilde{q}, r; P\kappa)\ .
\end{aligned}$$

Define an equivalence relation

$$\mathcal{R} = \{(s_1, s_3), (s_2, s_4)\} \cup \mathcal{I}$$

where $\mathcal{I}$ is the identity relation. We have $s_1 \xrightarrow{\tau} s_2$ and $s_3 \xrightarrow{\tau} s_4$ where $(s_2, s_4) \in \mathcal{R}$, and we have $s_2 \xrightarrow{\tau} s_5$ and $s_4 \xrightarrow{\tau} s_5$ where $(s_5, s_5) \in \mathcal{R}$. Therefore $\mathcal{R}$ is a probabilistic branching bisimulation. □

Restriction

The following lemma proves the soundness of the restriction laws R1–R3.

Lemma 5.8 (Restriction Laws)**.** *For any P, Q and c:*

1. $(\nu c)(P + Q) \leftrightarrows^c (\nu c)P + (\nu c)Q$,
2. $(\nu c)\alpha.P \leftrightarrows^c \mathbf{0}$ *if* $\alpha \in \{c?[\cdot], c![\widetilde{v}, \widetilde{q}]\}$,
3. $(\nu c)\alpha.P \leftrightarrows^c \alpha.(\nu c)P$ *if* $\alpha \notin \{c?[\cdot], c![\cdot]\}$.

Proof. 1. Define a relation $\mathcal{R}$ as the identity relation extended with the pairs $((\sigma; \omega; (\nu c)(P+Q)), (\sigma; \omega; (\nu c)P + (\nu c)Q))$. The transition $(\sigma; \omega; (\nu c)(P+Q)) \xrightarrow{\alpha} (\sigma'; \omega'; (\nu c)R)$ is derived by L-Res and L-Sum. Then applying these rules in the reverse order we can derive $(\sigma; \omega; (\nu c)P + (\nu c)Q) \xrightarrow{\alpha} (\sigma'; \omega'; (\nu c)R)$. Therefore we have $\mathcal{R}$ is a probabilistic branching bisimulation.

2. Define a relation $\mathcal{R}$ as the pair $((\sigma; \omega; (\nu c)\alpha.P), (\sigma; \omega; \mathbf{0}))$. If $\alpha \in \{c?[\cdot], c![\widetilde{v}, \widetilde{q}]\}$ then $(\sigma; \omega; (\nu c)\alpha.P)$ has no transitions. The configuration $(\sigma; \omega; \mathbf{0})$ has no transitions by definition, hence $\mathcal{R}$ is a bisimulation.

3. Define a relation $\mathcal{R}$ as the identity relation extended with

$$\{((\sigma; \omega; \alpha.(\nu c)P), (\sigma; \omega; (\nu c)\alpha.P)) \mid \alpha \notin \{c?[\cdot], c![\cdot]\}\}\ .$$

   We consider the derivation of the transition $(\sigma; \omega; \alpha.(\nu c)P) \xrightarrow{\beta} s'$. If the derivation is by L-Expr then $s' = (\sigma'; \omega; \alpha'.(\nu c)P)$ and by L-Res and L-Expr we

have $(\sigma;\omega;(\nu c)\alpha.P) \xrightarrow{\beta} (\sigma';\omega;(\nu c)\alpha'.P) = t'$ and therefore $(s',t') \in \mathcal{R}$. If the derivation is by either L-IN, L-QBIT or L-ACT then $s' = (\sigma';\omega';(\nu c)P')$. Applying L-RES followed by the respective rule gives the transition

$$(\sigma;\omega;(\nu c)\alpha.P) \xrightarrow{\beta} (\sigma';\omega';(\nu c)P' = t'$$

and therefore $(s',t') \in \mathcal{R}$. If the derivation is by L-OUT then $s' = \boxplus_{m\in M}\ p_m \bullet \oplus_{i\in I_m}\ g_i\ (\sigma'_i;\omega';\lambda\widetilde{x}.(\nu c)P;\widetilde{v}_i)$ and by L-RES and L-OUT we have the transition

$$\oplus_{i\in I}\ \ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}.(\nu c)\alpha.P;\widetilde{v}_i) \xrightarrow{\beta} \boxplus_{m\in M}\ \ p_m \bullet \oplus_{i\in I_m}\ \frac{g_i}{p_m}\ (\sigma'_i;\omega';\lambda\widetilde{x}.(\nu c)P;\widetilde{v}_i) = t'$$

and therefore $(s',t') \in \mathcal{R}$.

□

5.3 Expanding processes

In this section, we discuss the issues preventing the expansion law of classical process calculus from being adapted to CQP. We present the natural conversion of the expansion law from the π-calculus into CQP, and illustrate how such a law could be applied to quantum teleportation. In doing so, we show that the process *Teleport* is congruent to the flattened version (Equation 5.1, used as the starting point in Section 5.1), noting — informally — that in this particular case the equalities are correct.

The *Teleport* process consists of a preparation stage (the creation of the entangled qubits) followed by the parallel composition of processes *Alice* and *Bob*. In this particular protocol, there is no interleaving between the parallel components because Bob must wait until Alice sends her measurement values. Therefore, it is natural to expect that we can expand this parallel composition into a single sequential process; the *expansion law* is the conventional method for this.

5.3.1 CQP and The Expansion Law

Adapting the *expansion lemma* from the π-calculus (see [Sangiorgi and Walker 2001, Lemma 2.2.14]) results in the following proposed equality, expressed with respect to CQP.

If $M = \sum_{i=1}^{m} \alpha_i.P_i$ and $N = \sum_{j=1}^{n} \beta_j.Q_j$, then

$$M \parallel N \leftrightarrows^c \sum_{i=1}^{m} \alpha_i.(P_i \parallel N) + \sum_{j=1}^{n} \beta_j.(M \parallel Q_j) + \sum_{\alpha_i\ \mathbf{comp}\ \beta_j} \tau.R_{ij}$$

where α_i **comp** β_j if α_i is $c![\widetilde{e}]$ and β_j is $c?[\widetilde{x}]$ and $R_{ij} = P_i \parallel Q_j\{\widetilde{e}/\widetilde{x}\}$.

The expansion law serves to make every possible action explicit, by unfolding the parallel composition into a summation. The prefixes of the terms in the first two parts of this summation correspond to the capabilities of M and N respectively. The third part is a summation corresponding to the potential communications between M and N. The condition α_i **comp** β_j identifies *complementary* actions, that is when α_i is an output and β_j is a matching input.

Before applying this law to teleportation, we discuss three issues which prevent its application to general processes. First, there is no τ prefix in CQP, however this is easily overcome by replacing the τ by, for example, $\{\mathsf{unit}\}$ because the semantics of the $\{\mathsf{unit}\}$ prefix is analogous to the semantics of the τ prefix in the π-calculus.

The second issue is concerned with the evaluation of expressions in CQP. In contrast to the π-calculus, because CQP includes the syntax and semantics for the evaluation of expressions, a prefix does not necessarily correspond to a single action. The consequence of this is that the execution tree of the left hand process has more states than that of the right hand process. The process $c![\mathsf{measure}\ x].\mathbf{0} \parallel d![\mathsf{measure}\ y].\mathbf{0}$ provides a simple counter-example in which the expansion is not bisimilar; in the expansion, for example, the measurement of x would preclude the output on channel d from occurring before the output on channel c.

The third issue that we consider is concerned with the output of expressions. The term R_{ij}, which results from communication, features the substitution $\{\widetilde{e}/\widetilde{x}\}$ in order to transfer the expression from one process to another. The effects of transferring expressions in this way are multiple; evaluation of the expression is deferred since normally evaluation would occur before the communication; evaluation may occur multiple times if $\widetilde{x}$ appears more than once, which is a problem for non-idempotent expressions such as $\widetilde{y} *= U$; and typing is not preserved, for example, if the expression is $\mathsf{measure}\ x$ then the transfer may contradict the unique ownership theorem.

5.3.2 Expanding Teleportation

Despite the general incompatibility of the expansion law, the issues mentioned previously cause little concern in the case of teleportation. Primarily, this is because the teleportation process exhibits no branching, which is common to all of the aforementioned points. Although the analysis in the section is not rigorous, it provides useful insight into the role of equational reasoning, and also reinforces the potential existence of a suitable expansion law for CQP. This analysis also serves to illustrate the restriction laws given in Figure 5.4.

We begin by applying the expansion law to the parallel process *Alice* $\parallel$ *Bob*. Let

$Alice = c?[x].Alice'$ and $Bob = e?[r,s].Bob'$, then we have

$$Alice \parallel Bob = c?[x].(Alice' \parallel Bob) + e?[r,s].(Alice \parallel Bob') \ .$$

The first term corresponds to Alice's initial input action, the second term to Bob's input action. There are no cases in which they can communicate at this point, hence the final term is empty.

Restrictions

We know that Bob cannot start until Alice is ready to send the measurement results, therefore we expect that the second term, prefixed by Bob's input $e?[r,s]$, has no behaviour. Specifically, this inability to execute is a result of the restriction (νe) covering Alice and Bob. In Figure 5.4 we included the following rules for restriction:

$$(\nu c)(P+Q) = (\nu c)P + (\nu c)Q \qquad \text{(R1)}$$

$$(\nu c)\alpha.P = \mathbf{0} \qquad \text{if } \alpha \in \{c?[\cdot], c![\widetilde{v}, \widetilde{q}]\} \qquad \text{(R2)}$$

$$(\nu c)\alpha.P = \alpha.(\nu c)P \qquad \text{if } \alpha \notin \{c?[\cdot], c![\cdot]\} \qquad \text{(R3)}$$

These rules enable us to move restrictions in and out of summations, identify semantically null processes, and commute with prefixes. Using R1 we can move the restriction into the sum:

$$(\nu e)(c?[x].(Alice' \parallel Bob) + e?[r,s].(Alice \parallel Bob')) = \\ (\nu e)c?[x].(Alice' \parallel Bob) + (\nu e)e?[r,s].(Alice \parallel Bob') \ .$$

Now, we can apply R2 to the second term, identifying it as semantically null:

$$(\nu e)e?[r,s].(Alice \parallel Bob') = \mathbf{0} \ .$$

Combined with structural congruence, we can remove this term from the summation. Finally, R3 allows us to bring the input $c?[x]$ outside the restriction:

$$(\nu e)c?[x].(Alice' \parallel Bob) = c?[x].(\nu e)(Alice' \parallel Bob) \ .$$

We have now expanded part of the parallel composition and eliminated semantically null terms to give

$$(\nu e)(Alice \parallel Bob) = (\nu e)c?[x].(Alice' \parallel Bob) \ .$$

By applying this procedure iteratively, we can fully expand the process *Alice* up

to the output action on e. At this point we have

$$(\nu e)(\mathit{Alice} \parallel \mathit{Bob}) = \\ c?[x].\{x, z *\!\!= \mathsf{CNot}\}.\{x *\!\!= \mathsf{H}\}.(\nu e)(e![\mathsf{measure}\ z, \mathsf{measure}\ x].\mathbf{0} \parallel \mathit{Bob}) \ .$$

The next application of the expansion law involves the communication between *Alice* and *Bob*, giving

$$(e![\mathsf{measure}\ z, \mathsf{measure}\ x].\mathbf{0} \parallel \mathit{Bob}) = e![\mathsf{measure}\ z, \mathsf{measure}\ x].(\mathbf{0} \parallel \mathit{Bob}) \\ + e?[r, s].(e![\mathsf{measure}\ z, \mathsf{measure}\ x].\mathbf{0} \parallel \mathit{Bob}') \\ + \tau.(\mathbf{0} \parallel \mathit{Bob}\{\mathsf{measure}\ z, \mathsf{measure}\ x/r, s\}) \ .$$

In this case, we have complementary actions $e?[r, s]$ and $e![\mathsf{measure}\ z, \mathsf{measure}\ x]$ which result in a third term in the summation, representing the communication that can occur. By including the restriction, we are able to identify the first two terms as semantically null processes, thus

$$(\nu e)(e![\mathsf{measure}\ z, \mathsf{measure}\ x].\mathbf{0} \parallel \mathit{Bob}) = \mathbf{0} + \mathbf{0} \\ + (\nu e)\tau.(\mathbf{0} \parallel \mathit{Bob}\{\mathsf{measure}\ z, \mathsf{measure}\ x/r, s\}) \ .$$

The $\mathbf{0}$ processes can be removed using structural congruence, and through several iterations of R3 followed by the structural congruence rule $(\nu e)0 \equiv 0$, we have

$$(\nu e)(\mathit{Alice} \parallel \mathit{Bob}) = \\ c?[x].\{x, z *\!\!= \mathsf{CNot}\}.\{x *\!\!= \mathsf{H}\}.\tau.\mathit{Bob}\{\mathsf{measure}\ z, \mathsf{measure}\ x/r, s\} \ .$$

Aside from the additional τ which serves no purpose in this case, the process obtained is identical to the flattened version of teleportation given in Equation (5.1), having eliminated all parallel constructions. Although the measurement expressions $\mathsf{measure}\ x$ and $\mathsf{measure}\ z$ have been deferred as a result of the expansion, there is no adverse consequence; instead we obtain a process similar to the $\mathit{Teleport}_D$ process introduced in Section 3.3, which is based on the principle of deferred measurement.

5.4 Summary

In Chapter 4, we defined full probabilistic branching bisimilarity and proved that it is a congruence. In Section 4.4, we proved that the process *Teleport* is congruent to *QChannel* by defining a full probabilistic branching bisimulation. In this chapter, we developed an equational theory of full probabilistic branching bisimilarity in order to

enable the determination of equivalence through equational reasoning.

In Section 5.1, we used the quantum teleportation process to motivate a number of equational laws. In doing this, we demonstrated the application of equational reasoning to the analysis of a practical quantum process. Due to the difficulty of adapting the expansion law from classical process calculi, the analysis started with a flattened version of the *Teleport* process. In Section 5.3, we discussed the issues surrounding the use of the expansion law with CQP, and the potential use of an appropriate law to manipulate the parallel composition found in the original *Teleport* process.

In Section 5.2, we proved the soundness of the equational laws presented in this chapter, as summarised in Figure 5.4.

6

A Combined Approach to Quantum Verification

In the previous chapters, we described the use of process calculus for the verification of quantum protocols. Process calculus is an established and successful field in classical computer science, providing the framework for an algebraic approach to system analysis. We have used manual proof techniques to verify properties of simple quantum systems, although, while this is an effective approach, it can become unweildy for analysing more complex systems.

Automated techniques are often considered a better approach to verification, not only due to the offloading of work to a machine, but also because the precise nature of a machine removes the possibility of incorrect proofs. *Model–checking* [Clarke et al. 2000] is one such automated technique for formal verification and works by performing an exhaustive search on the state space of the system in question. Properties for system correctness are usually specified using temporal logic formulae, which can be checked against the resulting state space. Model–checking differs from testing in its exhaustive nature. Testing is normally based on a sample of conditions which are meant to represent a range of possible scenarios. This is often limited by the imagination of the designer, a limitation that is avoided in model–checking.

Classical model checking tools are not fully adequate for the analysis of quantum systems, in much the same way that classical process calculi struggle. The ability to represent quantum information and its properties is required for analysing all but the basic properties. Previous attempts at using existing tools include the modelling and verification of the BB84 key distribution protocol [Bennett and Brassard 1984] with CCS and the Concurrency Workbench of the New Century (CWB-NC) tool by Nagarajan and Gay [2002], and the use of the PRISM probabilistic model checker [Kwiatkowska et al. 2001] by Gay et al. [2005] to check a selection of quantum protocols

including quantum teleportation [Bennett et al. 1993] and quantum error correction. These efforts provided the foundation for both CQP and the Quantum Model Checker (QMC) tool.

There are many software tools and libraries that have been developed for the simulation of quantum systems on classical computers (see [Glendinning 2010] for a partial list), however the Quantum Model Checker (QMC) [Gay et al. 2007, 2008; Papanikolaou 2009] is a tool that goes beyind simulation and is designed for verification. Simulation plays a central role in model checking, although it is not possible to simulate quantum computations efficiently on a classical computer; indeed this is one of the major advantages promised by quantum computers. This inefficiency is compounded by the fact that model checking is computationally expensive, since all possible executions of a system must be explored.

The *stabilizer formalism* is a sub-class of quantum computation which include the Pauli, Hadamard, controlled-NOT and phase operators. This class of computation is particularly interesting because, according to the Gottessman-Knill Theorem, it can be simulated on a classical computer in polynomial time and space. The advantage of the stabilizer formalism lies within the representation of the quantum state; instead of a state vector representation, the quantum state can be specified by the operators which are *stabilizers* (a group theoretic concept in which the stabilizers of a group are those operators which map the group to itself). The obvious disadvantage of the stabilizer formalism is that it falls short of the full power of quantum computation needed to implement quantum algorithms.

QMC was designed to take advantage of the efficient representation and simulation algorithms offered by the stabilizer formalism, and it has been tested with a number of small but practical case studies. However, the stabilizer formalism is too restrictive in the long term, since the verification of larger systems and complex protocols with security requirements requires support for arbitrary quantum operators. The security of quantum coin flipping protocols [Berlín et al. 2008] is an example where there are known attacks using non-stabilizer quantum states, while the normal protocol operates within the scope of the formalism.

In this chapter we define a translation from CQP to the modelling langauge of QMC. This will enable the expressive power of CQP, combined with the process equivalence in Chapter 4, to be used in conjunction with the automated approach offered by QMC. The semantic correctness of the translation is proved, in order to ensure that translated specifications describe the same system. There is considerable scope for extending both CQP and QMC, and this chapter will provide the technical underpinning to develop this translation alongside the languages.

This chapter is based on the CQP language and reduction semantics of [Gay and Nagarajan 2006], and does not consider the modifications featured in Chapters

3 and 4. In particular, QMC is designed to model closed systems and therefore the ability to model external interactions, as introduced in the previous chapters, is not required. Indeed, if the semantics of external interactions is removed from Chapters 3 and 4, then there are only minor differences between these semantics and the reduction semantics in [Gay and Nagarajan 2006]. However, this does highlight a major difference between the applications of bisimilarity and model-checking; in the latter case, external entities must be specified explicitly in order to define a closed system.

6.1 Modelling Quantum Protocols in QMC

In this section we introduce the *Quantum Model Checker (QMC)* and its use in the verification of quantum protocols. QMC is a software tool developed by Gay et al. [2008], that automatically explores all possible behaviours arising from a protocol model, and enables logic properties expressed with *Quantum Computation Tree Logic (QCTL)* [Baltazar et al. 2008] to be checked over the resulting structure.

In QMC, a protocol model consists of one or more processes, where each process describes a series of commands. The commands of each process are interleaved to model concurrent execution. Non-determinism arises through interleaving, selection constructs and measurement, which is resolved to produce an *execution tree* for the modelled system.

Each node in the execution tree is represented by a *configuration*. This a tuple $(P, \kappa, \Sigma, |\psi\rangle)$, where P is the abstract representation of the program, κ is the global store, Σ is the set of local stores, each corresponding to a process, and $|\psi\rangle$ is the quantum state. The inclusion of global and local stores is a reflection of the computational nature of QMC, as opposed to the algebraic nature of CQP in which values are contained within the process.

The internal representation of the quantum state is of particular interest; rather than storing the *state vector* representation of $|\psi\rangle$ (which grows exponentially in size with the number of qubits), the *stabilizer array representation* is used in QMC. This is a binary representation of the set of Pauli operators that stabilize (or fix) $|\psi\rangle$. This representation results in significant computational benefits in term of both space and time when simulating protocols. This is because the size of the representation grows polynomially with the number of qubits, and because there are polynomial-time algorithms for the simulation of stabilizer circuits [Aaronson and Gottesman 2004].

6.1.1 Syntax

Models are written in an imperative-style concurrent specification language that has been developed for QMC. The syntax of the language is defined by the grammar in

$$
\begin{array}{rcl}
t & ::= & \textbf{integer} \mid \textbf{bool} \mid \textbf{real} \mid \textbf{qubit} \mid \textbf{channel of}\ t \\
e & ::= & n \mid r \mid x \mid e_1 + e_2 \mid e_1 - e_2 \mid e_1 * e_2 \mid e_1/e_2 \\
 & \mid & \textbf{true} \mid \textbf{false} \mid \texttt{not}\ e \mid e_1\ \texttt{and}\ e_2 \mid e_1\ \texttt{or}\ e_2 \\
 & \mid & e_1 = e_2 \mid e_1 < e_2 \mid e_1 > e_2 \mid \texttt{meas}\ e \mid \texttt{newqubit} \\
S & ::= & e \mid x := e \mid x_1!x_2 \mid x_1?x_2 \mid \textbf{cnot}\ x_1\ x_2 \mid \textbf{had}\ x \\
 & \mid & \textbf{ph}\ x \mid \textbf{X}\ x \mid \textbf{Y}\ x \mid \textbf{Z}\ x \mid S_1; S_2 \\
H & ::= & \texttt{::}\ SH \mid \texttt{::}\ S \\
GC & ::= & \textbf{if}\ H\ \textbf{fi} \mid \textbf{do}\ H\ \textbf{od} \\
C & ::= & S; \mid GC \mid C_1\ C_2 \mid \epsilon \\
VD & ::= & \textbf{var}\ x : t; VD \mid \epsilon \\
P & ::= & \textbf{process}\ p\ VD\ \textbf{begin}\ C\ \textbf{end}\ P \mid \epsilon \\
M & ::= & \textbf{program}\ p\ VD\ \textbf{begin}\ P\ \textbf{end}
\end{array}
$$

Figure 6.1. QMC Concrete Syntax

Figure 6.1 (from [Papanikolaou 2008]). Expressions e consist of names, values, arithmetic operators, boolean operators, quantum measurement and initialisation of new qubits. Statements S consist of expressions, assignment, output, input, quantum operations (limited to operators in the stabilizer formalism) and sequential composition. Options H allow a choice between one or more statements; these are used in guarded commands GC, which consist of the **if** and **do** constructs. Commands C consist of statements, guarded commands, or a sequence of commands. Variable declarations VD allow a (possibly empty) sequence of declarations. Processes P are a (possibly empty) sequence of processes (these processes are executed concurrently and not sequentially), each containing variable declarations and commands. A program M is a single construct containing (global) variable declarations and processes.

The syntax is best demonstrated by an example. The program in Figure 6.2 illustrates the modelling of quantum teleportation in QMC. We assume the state to be "teleported" is $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

The language allows for global variables (such as `e1`, `e2`), typed communication channels (such as `ch`) which are always global, and local (private) variables for each process (such as `a`,`b`,`c`,`d`,`q`). Communication is asynchronous, with executability rules restricting the way in which the interleaving of process is performed. For instance, the process `Bob` cannot start unless channel `ch` is filled with a value.

6.1.2 Verification with QMC

There are several properties of quantum protocols that we are interested in reasoning about. In particular, we are interested in properties of the quantum state, such as

```
program Teleport;
var e1,e2:qubit; ch:channel of integer;
process Alice;
var q:qubit; a,b:integer;
begin
 q := newqubit; had q;
 e1 := newqubit; e2 := newqubit;
 had e1; cnot e1 e2;
 cnot q e1; had q;
 a := meas q;
 b := meas e1;
 ch!a; ch!b;
end;
process Bob;
var c,d: integer;
begin
 ch?c; ch?d;
 if
 :: ((c=1) and (d=0)) -> X q; break;
 :: ((c=0) and (d=1)) -> Z q; break;
 :: ((c=1) and (d=1)) -> X q; Z q; break;
 :: ((c=0) and (d=0)) -> break;
 fi
end;
endprogram.
```

Figure 6.2. Quantum teleportation modelled in QMC.

$$
\begin{aligned}
\alpha &::= \bot \mid \mathsf{qb} \mid \alpha \Rightarrow \alpha \mid \alpha \vee \alpha \mid \alpha \wedge \alpha \\
t &::= x \mid (t+t) \mid (tt) \mid \mathrm{Re}(|\top\rangle_A) \mid \mathrm{Im}(|\top\rangle_A) \mid \int \phi \\
\gamma &::= (t \leq t) \mid \bot \mid (\alpha \sqsupset \alpha) \mid (\alpha \curlyvee \alpha) \mid (\alpha \curlywedge \alpha) \mid [\mathsf{qb}_i, \mathsf{qb}_j, \ldots] \\
\theta &::= \gamma \mid \theta \sqsupset \theta \mid (\mathsf{EX}\theta) \mid ([\theta \ \mathsf{EU} \ \theta]) \mid (\mathsf{AF}\theta)
\end{aligned}
$$

Figure 6.3. Syntax of QCTL.

which qubits are "active" in a given state and which qubits are entangled with the rest of the system. The evolution of classical values is also important, including the possible outcomes of measurements. These properties can be expressed for QMC models using QCTL [Baltazar et al. 2008].

QCTL adds the usual temporal connectives (AX, EF, EU) of *Computation Tree Logic (CTL)* [Emerson 1990] to the propositional logic EQPL [Mateus and Sernadas 2006]. The meaning of formulae in *Exogenous Quantum Propositional Logic (EQPL)* is expressed in terms of valuations, which are truth-value assignments for the symbols $\mathsf{qb}_0, \mathsf{qb}_1, \ldots, \mathsf{qb}_n$ corresponding to each qubit in the system. For example, the quantum state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle$ is understood as a pair of valuations (v_1, v_2) for a 2-qubit system such that $v_1(\mathsf{qb}_0) = 0$, $v_1(\mathsf{qb}_1) = 0$, $v_2(\mathsf{qb}_0) = 1$, $v_2(\mathsf{qb}_1) = 1$.

The formulae accepted by the QMC tool for verification allow the user to reason about the state of individual qubits, and involve usual logical connectives such as negation and implication. The syntax of QCTL is given in Figure 6.3 (from [Baltazar et al. 2008]):

There are two levels of formulae: classical formulae (α), which hold only if all valuations in a state satisfy them, and quantum formulae (γ), which are essentially logical combinations of classical formulae. For instance, the quantum conjunction in the formula $\phi_1 \curlywedge \phi_2$ is only satisfied if both the classical formulae ϕ_1 and ϕ_2 are satisfied in the current state. A particularly distinctive type of quantum formula is of the form $[Q]$, where Q is a list of qubit variables $\mathsf{qb}_i, \mathsf{qb}_j, \ldots$; this type of formula is satisfied only if the qubits listed are separable from all other qubits in the system.

Example of a Property for Verification

We have considered the correctness requirement for the teleportation in the process calculus setting, that is, bisimilarity to a quantum channel. We can express a similar requirement in the context of a QMC model; at the end of the protocol, the third qubit will be in the same state as the first qubit was to begin with, and this qubit will be disentangled from the rest of the system. We can express this requirement, for the case where the input is the quantum state $|0\rangle$, in the input language of QMC

using the specification

```
finalstateproperty ([q2]) #/\ (!q2);
```

which corresponds to the EQPL formula $[q_2] \curlywedge (\neg q_2)$. The first part of the formula asserts that the last qubit (`q2`) is disentangled from the rest of the system, while the second part asserts that the current valuation assigns to this qubit a value of 0. The entire formula is true if both parts are true, indicated by the connective of quantum conjunction (we represent $\curlywedge$ in ASCII form by `#/\`). We can also use a temporal formula:

```
property true EU (([q2]) #/\ (!q2));
```

6.2 Translation

In this section we define a translation from CQP processes to QMC programs. This translation has been developed partly in tandem with QMC, and is based on the formal syntax and semantics in the unpublished report [Papanikolaou 2008] and the updated semantics in [Papanikolaou 2009]. The translation has the potential to be adapted following any future developments in CQP and QMC.

There are several differences between the languages that result in limitations or special treatment in the translation. The most significant, due to the inability to model universal quantum computation, is the restriction to processes that fall within the stabilizer formalism. Other issues, which we discuss in more detail in the following sections, include the removal of channel mobility, translating from polyadic to monadic channels, and allowing only single qubit measurements.

To simplify the presentation, we also require that all variable names are unique among all CQP processes; this can be achieved by alpha conversion if necessary. As a result we are able to define all variables globally when translated to QMC without risk of collision.

6.2.1 Translation Functions

In this section, we define a function $\textsc{Tprog}[\![]\!] : \mathcal{P}_C \longrightarrow \mathcal{P}_Q$ where $\mathcal{P}_C$ and $\mathcal{P}_Q$ are the set of CQP processes and QMC programs respectively. We follow a similar approach to Nielson and Nielson [1999]; the translation is defined in several steps on the structural elements of CQP. We will introduce functions to translate processes in parallel contexts, expressions, types and values.

A QMC program consists of one or more named processes. Although the formal syntax of CQP does not feature named processes, we choose to use them as standard

$$
\begin{array}{rcl}
T & ::= & \mathsf{Int} \mid \mathsf{Unit} \mid \mathsf{Qbit} \mid \widehat{\ }[\tilde{T}] \mid \mathsf{Op(1)} \mid \mathsf{Op(2)} \mid \ldots \\
e & ::= & v \mid \mathsf{measure}\ \tilde{e} \mid \tilde{e} *\!= e^e \mid e + e \\
v & ::= & x \mid \mathsf{0} \mid \mathsf{1} \mid \cdots \mid \mathsf{unit} \mid \mathsf{H} \mid \mathsf{X} \mid \cdots \\
P & ::= & \mathbf{0} \mid (ProcName(\tilde{x}) \parallel ProcName(\tilde{x})) \mid e?[\tilde{x} : \tilde{T}].P \mid e![\tilde{e}].P \\
 & \mid & \{e\}.P \mid (\nu x{:}\widehat{\ }[T])P \mid (\mathsf{qbit}\ x)P \\
D & ::= & ProcName(\tilde{x} : \tilde{T}) = P \\
C & ::= & \tilde{D}
\end{array}
$$

Figure 6.4. CQP syntax with named processes.

(as, for example, the processes *Teleport*, *Alice* and *Bob* in Figure 6.10) instead of introducing arbitrary names as part of the translation. In this approach, the parallel composition primitive must use process names instead of processes. For example, the process $P.(Q \parallel R)$ would be represented by the named processes

$$
\begin{array}{rcl}
Process1 & = & P.(Process2 \parallel Process3) \\
Process2 & = & Q \\
Process3 & = & R\ .
\end{array}
$$

Formally, for this translation we define additional non-terminals, D for process definitions and C for a collection of process definitions, to the CQP syntax in order to introduce named processes. Processes P are also modified such that parallel process constructs are written as calls to named processes. This extended grammar is shown in Figure 6.4.

These new syntactic elements do not change the language, however we must remember that each named process may only appear once. We shall refer to the collection of process definitions as a CQP *program*, in order to distinguish between the collection and the individual processes.

We now define a series of transcription functions from the syntactic elements of CQP to the corresponding QMC syntax. These are: TPROG[[]] for the program; TPROC[[]] for processes; TEXPR[[]] for expressions; TVAL[[]] for values; and TTYPE[[]] for types.

The Program

The complete CQP program (C), which consists of the list of named processes (D_1, ..., D_n), must be rewritten to a QMC program. This top-level transcription is performed using the function TPROG[[]], defined in Figure 6.5. The resulting QMC program is encapsulated in a **`program`** block and contains global variable declarations

$$
\begin{aligned}
&\textsc{Tprog}[\![C = D_1, \ldots, D_n]\!] = \\
&\qquad \texttt{program Translated; } gVars(C)\texttt{;} \\
&\qquad \textsc{Tprog}[\![D_1]\!] \cdots \textsc{Tprog}[\![D_n]\!] \\
&\qquad \texttt{endprogram} \\
&\textsc{Tprog}[\![\mathsf{ProcName}(\widetilde{x}{:}\widetilde{T}) = P]\!] = \\
&\qquad \texttt{process ProcName; } lVars(P) \\
&\qquad \texttt{begin } isInvoked(\mathsf{ProcName}) \\
&\qquad \textsc{Tproc}[\![P]\!] \\
&\qquad \texttt{end}
\end{aligned}
$$

Figure 6.5. Translation of programs.

followed by a list of process blocks. Each process block corresponds to a named CQP process, and these are translated in turn using Tprog[[]]. We describe the generation of global and local variables declarations (*gVars* and *lVars*) later.

Each QMC process declaration defines a single process containing local variable declarations and process body, structured with the **process**, **begin** and **end** keywords. Given a single CQP process definition as input, Tprog[[]] rewrites this definition into a QMC process declaration.

For processes that are invoked from (that is, nested within) other processes the function *isInvoked* (Definition 6.2) inserts a receive statement (`Proc_ctrl?signal;`, where *Proc* is the process name) that is used to signal the start of execution. For non-invoked processes the function *isInvoked* produces no output (the empty string ε). The set of invoked processes $P_{in}(C)$ (Definition 6.1) is determined by analysis of the structure of the CQP program C.

Definition 6.1 (P_{in}: Set of Invoked Processes)**.** $P_{in}(C)$ gives the set of names of invoked processes in the CQP program C. Let $C = \widetilde{D}$, then this set is defined on the

structure of C as follows

$$\begin{aligned}
P_{in}(D_1, \ldots, D_n) &= \bigcup\nolimits_i P_{in}(D_i) \\
P_{in}(ProcName(\tilde{x} : \tilde{T}) = P) &= P_{in}(P) \\
P_{in}(\mathbf{0}) &= \emptyset \\
P_{in}((P1(\tilde{x}) \parallel P2(\tilde{y}))) &= \{P1, P2\} \\
P_{in}(e?[\tilde{e}].P) &= P_{in}(P) \\
P_{in}(e![\tilde{e}].P) &= P_{in}(P) \\
P_{in}(\{e\}.P) &= P_{in}(P) \\
P_{in}((\nu x{:}\hat{}[\tilde{T}])P) &= P_{in}(P) \\
P_{in}((\mathsf{qbit}\ x)P) &= P_{in}(P)
\end{aligned}$$

This set is used in the following formal definition of *isInvoked*.

Definition 6.2 (*isInvoked*: Invoked Process Function)**.** *isInvoked* is a function from process names to QMC statements, where

$$isInvoked(\mathsf{Proc}) \mapsto \begin{cases} \texttt{Proc\_ctrl?signal;} & \text{if } \mathsf{Proc} \in P_{in}(C) \\ \varepsilon & \text{otherwise.} \end{cases}$$

After any control statement has been added, the CQP process body is translated by the function TPROC⟦⟧.

Processes

The function TPROC⟦⟧, for translating the process body, is defined in Figure 6.6. The **0** process, denoting inaction, is rewritten to an empty string. The "invocation" of parallel processes $(P1(\widetilde{x}) \parallel P2(\widetilde{y}))$ is achieved through signalling; as described in the previous section, processes that are invoked will await a signal before proceeding (determined by the function *isInvoked*) hence sending this signal allows the invoked process to begin execution.

CQP and QMC use different models of communication; in the former, communication is synchronous, thus an input and output action must execute as one step. In contrast, communication in QMC is asynchronous, therefore an output action occurs strictly, but not necessarily immediately, before a corresponding input action. We simulate synchronous communication in QMC by requiring the sending process to wait for an acknowledgement from the receiving process. Thus, each CQP output action will be followed by an input action (TEXPR⟦e⟧`_ack?ack`) when translated, and similarly an output action (TEXPR⟦e⟧`_ack!ack`) will follow each CQP input action.

$$
\begin{aligned}
&\textsc{Tproc}[\![\mathbf{0}]\!] = \varepsilon \\
&\textsc{Tproc}[\![(P1(\tilde{x}) \parallel P2(\tilde{y}))]\!] = \texttt{P1\_ctrl!signal; P2\_ctrl!signal;} \\
&\textsc{Tproc}[\![e?[x_1 : T_1, \cdots, x_n : T_n].P]\!] = \\
&\qquad \textsc{Texpr}[\![e]\!]\texttt{1?}\textsc{Tval}[\![x_1]\!] \cdots \textsc{Texpr}[\![e]\!]\texttt{n?}\textsc{Tval}[\![x_n]\!] \\
&\qquad \textsc{Texpr}[\![e]\!]\texttt{\_ack!ack;}\textsc{Tproc}[\![P]\!] \\
&\textsc{Tproc}[\![e![e_1, \cdots, e_n].P]\!] = \texttt{e\_1:=}\textsc{Texpr}[\![e_1]\!]\texttt{; } \ldots \texttt{e\_n:=}\textsc{Texpr}[\![e_n]\!]\texttt{;} \\
&\qquad \textsc{Texpr}[\![e]\!]\texttt{1!e\_1; } \ldots \textsc{Texpr}[\![e]\!]\texttt{n!e\_n;} \\
&\qquad \textsc{Texpr}[\![e]\!]\texttt{\_ack?ack;}\textsc{Tproc}[\![P]\!] \\
&\textsc{Tproc}[\![\{e\}.P]\!] = \textsc{Texpr}[\![e]\!]\ \textsc{Tproc}[\![P]\!] \\
&\textsc{Tproc}[\![(\nu x{:}\hat{}[T])P]\!] = \textsc{Tproc}[\![P]\!] \\
&\textsc{Tproc}[\![(\mathsf{qbit}\ x)P]\!] = \texttt{x := newqubit;}\textsc{Tproc}[\![P]\!]
\end{aligned}
$$

Figure 6.6. Translation of processes.

The communication models also differ in the fact that CQP channels are polyadic (allowing multiple subjects), whereas in QMC channels are monadic (having a single subject), hence it is necessary to separate CQP communication actions into multiple QMC actions. Because channels are typed in both languages, we cannot use a single channel name for all resulting actions. Instead, we introduce distinct channel names for each action. For example, a CQP channel c of arity n will correspond to channels $c_1, \ldots, c_n$ in QMC.

The use of distinct names also serves to prevent cross-talk and resulting deadlock. Consider, for example, two processes which can send on the same channel:

$$c![11, \ldots, 1n].\mathbf{0} \parallel c![21, \ldots, 2n].\mathbf{0}\ .$$

If, in the translation, we use the same channel name for each action, we would get

$$\texttt{c!11; } \ldots \texttt{c!1n; } \parallel \texttt{c!21; } \ldots, \texttt{c!2n; }\ .$$

This could result in an output sequence such as $11, 21, 22, 23, 12, \ldots$, in which values appear from both processes.

Furthermore, QMC allows only variable names as the subject in output actions, hence it is necessary to assign any value or expression to be sent to a fresh variable before sending. For example, the CQP action $ch![x, 3]$ is translated to

```
ch_1 := x; ch1!ch_1; ch_2 := 3; ch2!ch_2; ch_ack?ack;
```

Although it is not strictly necessary to make the assignment for x in this translation,

$\textsc{Texpr}[\![v]\!] = \textsc{Tval}[\![v]\!]$

$\textsc{Texpr}[\![\mathsf{measure}\ e_1, \ldots, e_n]\!] = \texttt{meas}\ \textsc{Texpr}[\![e_1]\!], \ldots, \textsc{Texpr}[\![e_n]\!]$

$\textsc{Texpr}[\![e_1, \ldots, e_n \mathrel{*}= e]\!] = \textsc{Texpr}[\![e]\!]\ \textsc{Texpr}[\![e_1]\!] \ldots \textsc{Texpr}[\![e_n]\!]$

$\textsc{Texpr}[\![e_1, \ldots, e_n \mathrel{*}= e^f]\!] = \textsc{Texpr}[\![e]\!]\texttt{\_cond := }\textsc{Texpr}[\![f]\!]$

```
if
::(TEXPR[[e]]_cond = 1) -> TEXPR[[e_1, ..., e_n *= e]] break;
::(TEXPR[[e]]_cond = 0) -> break;
fi
```

$\textsc{Texpr}[\![e + e]\!] = \textsc{Texpr}[\![e]\!]\texttt{+}\textsc{Texpr}[\![e]\!]$

Figure 6.7. Translation of expressions.

this convention is extended to variables for the purpose of generalisation.

The creation of a new channel $((\nu c{:}\widetilde{[T]}))$ has no corresponding statement in the QMC process body, however these constructs will be used later by the function an_G to collect channel and qubit declarations. In contrast, new qubit declarations $((\mathsf{qbit}\ q))$ have a corresponding initialisation statement in the QMC process body.

Expressions

CQP expressions consist of values, quantum measurements, quantum operations, and arithmetic expressions. We define the function $\textsc{Texpr}[\![]\!]$ in Figure 6.7 for the translation of expressions. Values v are translated by $\textsc{Tval}[\![]\!]$ (defined later). Quantum measurements on multiple qubits are possible in both languages, however they follow different conventions for the assignment of values to outcomes (decimal values $0, 1, 2, 3$ in CQP as opposed to bit tuples $(0,0), (0,1), (1,0), (1,1)$ in QMC). We therefore restrict the translation to single qubit measurements (on which the resulting values correspond) and note that this does not impact expressiveness.

There are two cases for quantum operators; conditional and non-conditional. In general, quantum operations are transcribed with the quantum operator first, followed by a comma separated list of qubit names. Controlled operations U^e are implemented using an **`if`** construct, preceded by an assignment to allow for the evaluation of an arbitrary expression. We only consider bit values (generally resulting from single qubit measurements) for controlled operations, although this could be extended to allow any integer using the fact that $U^4 = I$ for all operators in the stabilizer formalism.

Addition is the only arithmetic operator formally defined in CQP, however the translation is easily extended to other arithmetic expressions.

v	TVAL$[\![v]\!]$
$x, q, c, \ldots$	`x, q, c, ...`
$0, 1, \ldots$	`0, 1,...`
X, Y, Z	**`X`**, **`Y`**, **`Z`**
H, CNot, Ph	**`had`**, **`cnot`**, **`ph`**

T	TTYPE$[\![T]\!]$
Int	**`integer`**
Qbit	**`qubit`**
$\widehat{\ }[T]$	**`channel of`** TTYPE$[\![T]\!]$

Figure 6.8. Translation of values and types.

Values and Types

The functions TVAL$[\![]\!]$ and TTYPE$[\![]\!]$ defined in Figure 6.8 are used for the translation of values and types respectively. Variable names and literal values are left unchanged, while quantum operators are mapped to their QMC equivalents as per the definition. There is no translation for arbitrary quantum operators since only the Clifford operators are supported by QMC. The types Unit, Op(1),... corresponding to arbitrary operator types need not be translated since they are not used in QMC programs. Channel types $\widehat{\ }[\tilde{T}]$ make a recursive call to translate the component types. Since we don't allow channel mobility in the translation, declarations which are not allowed in QMC such as **`channel of channel of`** `T` are excluded.

Variable Declarations

QMC requires variables to be explicitly declared prior to use, either in a local process scope or globally. We take the approach of placing all variables in the global scope to avoid issues arising from the implementation of invocation. In particular, since channel names cannot be sent over QMC channels whereas other variables can, it is not possible to simulate inheritance of channel names. Instead we state the requirement that variable names must be unique, hence scoping in QMC will have no effect.

The functions *gVars* and *lVars* are used by TPROG$[\![]\!]$ in the translation of programs to provide global and local variable declarations respectively for the QMC program. *gVars* inspects the CQP program for binding operators including $c?[\widetilde{x}{:}\widetilde{T}]$ (gives declarations for each x_i), $(\nu x : \widehat{\ }[\widetilde{T}])$ (declares a series of channels x_i of type T_i), and (qbit x) (declares x as a qubit variable). Signalling channels for each process (`procname_ctrl:` **`channel of integer;`**) are also declared regardless of which processes will use them. Finally, *gVars* adds declarations for `signal` and `ack` which may be used for signalling and acknowledgments respectively.

Since we have chosen to declare all variables globally, the only use for *lVars* is to generate declarations for the intermediate assignments that arise from outputs and conditional unitary operations. An output $c![\widetilde{v}]$ will results in declarations `c_i: T_i` where T_i are the types associated with channel c, and an operator $\widetilde{q} *= U^e$ will result in a declaration `e_cond: T` where T is the type of expression e.

Formally, we define functions an_G (Definition 6.3) and an_L (Definition 6.4) to determine the variables of programs and single processes respectively.

Definition 6.3 (an_G: Global variable analysis)**.** Let Id denote a set of QMC identifiers and T a set of QMC types. Furthermore, let $IT = Id \times T$. We define a function $an_G : \mathcal{P}_C \to 2^{IT}$ inductively as:

$$\begin{aligned}
an_G(D_1 \dots D_n) &= \{(\texttt{signal}, \textbf{integer}), (\texttt{ack}, \textbf{integer})\} \\
&\quad \cup \textstyle\bigcup_i an_G(D_i) \\
an_G(Proc(\widetilde{x}{:}\widetilde{T}) = P) &= \{(\texttt{Proc_ctrl}, \textbf{channel of integer})\} \\
&\quad \cup an_G(P) \\
an_G(\mathbf{0}) &= \emptyset \\
an_G((P_1(\tilde{x}) \parallel P_2(\tilde{x}))) &= \emptyset \\
an_G(e?[e_1 : T_1, \dots, e_n : T_n].P) &= an_G(P) \cup \textstyle\bigcup_i \{(\texttt{e}i, \textsc{Type}[\![\widehat{\ }[T_i]]\!])\} \\
&\quad \cup \{(\texttt{e_ack}, \textbf{channel of integer})\} \\
an_G(e![\tilde{e}].P) &= an_G(P) \\
an_G(\{e\}.P) &= an_G(P) \\
an_G((\nu x{:}\widehat{\ }[T_1, \dots, T_n])P) &= an_G(P) \cup \textstyle\bigcup_i \{(\texttt{x}i, \textsc{Type}[\![\widehat{\ }[T_i]]\!])\} \\
an_G((\mathsf{qbit}\ \ x)P) &= an_G(P) \cup \{(q, \texttt{qbit})\}
\end{aligned}$$

Definition 6.4 (an_L: Local variable analysis)**.** We define a partial function $an_L : \mathcal{P}_C \rightharpoonup 2^{IT}$ inductively as:

$$\begin{aligned}
an_L(Proc(\tilde{x} : \tilde{T}) = P) &= an_L(P) \\
an_L(\mathbf{0}) &= \emptyset \\
an_L((P_1(\tilde{x}) \parallel P_2(\tilde{x}))) &= \emptyset \\
an_L(e?[\tilde{e} : \tilde{T}].P) &= an_L(P) \\
an_L(e![e_1, \dots, e_n].P) &= \{(\texttt{e_}i, \textsc{Type}[\![T_i]\!]) \mid 1 \leq i \leq n, \vdash e_i : T_i\} \\
&\quad \cup an_L(P) \\
an_L(\{e\}.P) &= an_L(P) \\
an_L(\{\widetilde{e} *{=} e^f\}.P) &= an_L(P) \cup \{(\texttt{e_cond}, \textsc{Type}[\![T]\!]) \mid \vdash f : T\} \\
an_L((\nu x{:}\widehat{\ }[T])P) &= an_L(P) \\
an_L((\mathsf{qbit}\ \ x)P) &= an_L(P)
\end{aligned}$$

We now define a function *vars* which takes a set of pairs (x, T) resulting from an_L and an_G, and produces a QMC variable declaration statement:

$$vars(\{(x_1 : T_1), \dots, (x_n : T_n)\}) = \texttt{x\_1: T\_1; \dots x\_n: T\_n;}$$

Then, for a CQP program consisting of processes $\widetilde{D}$, we have

$$gVars(\widetilde{D}) = vars(an_G(\widetilde{D}))$$
$$lVars(D) = vars(an_L(D))$$

6.3 Examples

In this section, we demonstrate the translation through some examples.

6.3.1 Random Bit Generator

The example consists of a quantum random number generator R which sends a random bit on a channel. The process Q simply receives the generated bit and is included to demonstrate communication. Process P is the starting process that invokes the other two. The three process definitions are:

$$P = (\nu c{:}\widehat{\ }[Int])(Q(c) \parallel R(c))$$
$$Q(c{:}\widehat{\ }[Int]) = c?[r : Int].\mathbf{0}$$
$$R(c{:}\widehat{\ }[Int]) = (\mathsf{qbit}\ x)(\{x \mathrel{*}= \mathsf{H}\}.c![\mathsf{measure}\ x].\mathbf{0})$$

We begin the translation by constructing the sets of global and local variables from these process definitions. These are generated by the functions an_G and an_L.

$$an_G(P,Q,R) = \{(\texttt{signal}, \textbf{integer}), (\texttt{ack}, \textbf{integer}),$$
$$(\texttt{Q\_ctrl}, \textbf{channel of integer}), (\texttt{R\_ctrl}, \textbf{channel of integer}),$$
$$(c1, \textbf{channel of integer}), (\texttt{c\_ack}, \textbf{channel of integer}), (x, \textbf{qubit}), (r, \textbf{integer})\}$$

The local variables are empty for processes P and Q, while for R there is a variable `c_1` used prior to sending:

$$an_L(P) = \emptyset$$
$$an_L(Q) = \emptyset$$
$$an_L(R) = \{(\texttt{c\_1}, \textbf{integer})\}$$

We now proceed by using the translation function $\textsc{Tprog}[\![]\!]$ on the set of process definitions.

$$\begin{aligned}\textsc{Tprog}[\![P, Q, R]\!] = \ & \texttt{program Translated;} \\ & gVars(P, Q, R) \\ & \textsc{Tproc}[\![P]\!]\ \textsc{Tproc}[\![Q]\!]\ \textsc{Tproc}[\![R]\!] \\ & \texttt{endprogram}\end{aligned}$$

The translations of each process P, Q and R are considered separately before being substituted in the above program.

For the body of process P we have the following translation:

$$\textsc{Tproc}[\![(\nu c{:}\hat{}[Int])(Q(c) \parallel R(c))]\!] = \textsc{Tproc}[\![Q(c) \parallel R(c)]\!] = \texttt{Q\_ctrl!signal; R\_ctrl!signal;}$$

The set of local variables $an_L(P)$ is empty, therefore there are no local declarations. The process definition is hence translated by $\textsc{Tprog}[\![]\!]$ to the following:

$$\begin{aligned}\textsc{Tprog}[\![P = (\nu c{:}\hat{}[Int])(Q(c) \parallel R(c))]\!] = \ & \texttt{process P} \\ & \texttt{begin} \\ & \texttt{Q\_ctrl!signal;} \\ & \texttt{R\_ctrl!signal;} \\ & \texttt{end}\end{aligned}$$

We now translate the process Q. The set of local variables $an_L(Q)$ is empty, therefore $lVars(Q)$ results in an empty string. The process body, consisting of a single input action is translated as

$$\textsc{Tproc}[\![c?[r]]\!] = \texttt{c1?r; c\_ack!ack;}\ ,$$

thus we arrive at the following:

$$\begin{aligned}\textsc{Tprog}[\![Q(c{:}\hat{}[Int]) = c?[r].\mathbf{0}]\!] = \ & \texttt{process Q} \\ & \texttt{begin} \\ & \texttt{c1?r; c\_ack!ack;} \\ & \texttt{end}\end{aligned}$$

The most interesting part is the translation of process R using $\textsc{Tproc}[\![]\!]$.

$$\begin{aligned}
&\textsc{Tproc}[\![(\mathsf{qbit}\ x)\{x *\!= \mathsf{H}\}.c![\mathsf{measure}\ x].\mathbf{0}]\!] \\
&\quad = \texttt{x := newqubit;}\ \textsc{Tproc}[\![\{x *\!= \mathsf{H}\}.c![\mathsf{measure}\ x].\mathbf{0}]\!] \\
&\quad = \texttt{x := newqubit;}\ \textbf{had}\ \texttt{x;}\ \textsc{Tproc}[\![c![\mathsf{measure}\ x].\mathbf{0}]\!] \\
&\quad = \texttt{x := newqubit;}\ \textbf{had}\ \texttt{x; c_1 := meas x; c1!c_1; c_ack?ack;} \\
&\qquad \textsc{Tproc}[\![\mathbf{0}]\!] \\
&\quad = \texttt{x := newqubit;}\ \textbf{had}\ \texttt{x; c_1 := meas x; c1!c_1; c_ack?ack;}
\end{aligned}$$

Combining this with the local variable declarations for process R we have the following translation:

$\textsc{Tprog}[\![R(c{:}\widehat{\ }[Int]) = (\mathsf{qbit}\ x)\{x *\!= \mathsf{H}\}.d![\mathsf{measure}\ x].\mathbf{0}]\!] =$

```
process R
var c_1:integer;
begin
x := newqubit;
had x;
c_1 := meas x; c1!c_1; c_ack?ack;
end
```

These three process translations can now be substituted into $\textsc{Tprog}[\![P, Q, R]\!]$ to complete the translation. The result is the QMC program listed in Figure 6.9.

6.3.2 Quantum Teleportation

We now apply this translation to the teleportation process defined in Figure 6.10. We expect the result to resemble the QMC program in Figure 6.2; the result is shown in Figure 6.11.

Unsurprisingly, the programs are not identical since differences in the languages allow for alternate representations of various components. The first point to note is the introduction of the signalling channels `Teleport_ctrl`, `Alice_ctrl` and `Bob_ctrl`. Since the *Teleport* process is not nested, the corresponding control channel is declared but never used. Incidentally, the use of the control channel in *Bob* is superfluous in teleportation, because execution cannot start until a value is received from *Alice*.

Another change is the conditional applications of the unitary operators by Bob; these have been compounded into one **`if`** statement in the QMC specification, however the simplistic support for conditionals by CQP leads to multiple statements in the

$\textsc{Tprog}[\![P, Q, R]\!] =$

```
program Translated;
  var signal:integer, ack:integer,
      Q_ctrl:channel of integer, R_ctrl:channel of integer,
      c1:channel of integer, c_ack:channel of integer,
      x:qubit, r:integer;
process P
begin
  Q_ctrl!signal; R_ctrl!signal;
end

process Q
begin
  c1?r; c_ack!ack;
end

process R
  var c_1:integer;
begin
  x := newqubit;
  had x;
  c_1 := meas x; c1!c_1; c_ack?ack;
end
endprogram
```

Figure 6.9. Translation of a quantum random number generator

$$\begin{aligned}
\mathit{Teleport} &= (\mathsf{qbit}\ y, z)(\{z \mathrel{*}\!\!= \mathsf{H}\}.\{z, y \mathrel{*}\!\!= \mathsf{CNot}\}.(\nu e{:}\widehat{\ }[\mathsf{Int}, \mathsf{Int}])(\mathit{Alice}(e, z) \parallel \mathit{Bob}(e, y))) \\
\mathit{Alice}(e, z) &= (\mathsf{qbit}\ x).\{x \mathrel{*}\!\!= \mathsf{H}\}.\{z, x \mathrel{*}\!\!= \mathsf{CNot}\}.\{z \mathrel{*}\!\!= \mathsf{H}\}.e![\mathsf{measure}\ z, \mathsf{measure}\ x].\mathbf{0} \\
\mathit{Bob}(e, y) &= e?[r{:}\mathsf{Int}, s{:}\mathsf{Int}].\{y \mathrel{*}\!\!= X^r\}.\{y \mathrel{*}\!\!= Z^s\}.\mathbf{0}
\end{aligned}$$

Figure 6.10. Quantum teleportation modelled in CQP.

```
program Translated;
var x: qubit; y: qubit; z: qubit;
  e1: channel of integer; e2: channel of integer;
  e_ack: channel of integer;
  Teleport_ctrl: channel of integer;
  Alice_ctrl: channel of integer;
  Bob_ctrl: channel of integer;
  r: integer; s: integer;
  signal: integer; ack: integer;
process Teleport;
begin
  y := newqubit; z := newqubit;
  had z; cnot z y;
  Alice_ctrl!signal; Bob_ctrl!signal;
end;

process Alice;
var e_1: integer; e_2: integer;
begin
  Alice_ctrl?signal;
  x := newqubit; had x; cnot z x; had z;
  e_1 := meas x; e1!e_1; e_2 := meas z; e2!e_2;
  e_ack?ack;
end;

process Bob;
var X_cond: integer; Z_cond:integer;
begin
  Bob_ctrl?signal;
  e1?r; e2?s; e_ack!ack;
  X_cond := r;
  if
  :: (X_cond = 1) -> X y; break;
  :: (X_cond = 0) -> break;
  fi
  Z_cond := s;
  if
  :: (Z_cond = 1) -> Z y; break;
  :: (Z_cond = 0) -> break;
  fi
end;
endprogram
```

Figure 6.11. Translated version of quantum teleportation.

translation. We have introduced the assignments `X_cond := r` and `Z_cond := s`, although in this particular case, since r and s are variables, they are not necessary.

6.4 Correctness of the Translation

In Section 6.2, we defined a translation from CQP processes to QMC programs. We now present a proof of the semantic correctness of this translation.

In order to argue that a QMC program translated from CQP has the same meaning as the original CQP process, it is necessary to show that the semantics of CQP processes are preserved by the translation function – this is equivalent to saying that the result of translating a CQP process into QMC then executing it should be identical to executing it in CQP then translating the result to QMC [Nielson and Nielson 1999].

The operational semantics of both languages are defined in terms of single step transitions from one configuration to another. Configurations, which include state information alongside the process, are necessary because the behaviour is not fully determined by the process. In order to reason about the equivalence of execution, we must extend the syntactic translation to a translation of configurations. This extension is given by the function $\mathcal{T}$, defined in Section 6.4.1.

We prove the semantic correctness of the translation by induction on the sequence of transitions in the execution. In Section 6.4.2, we investigate the relationship between CQP transitions and QMC transitions, in order to formulate our inductive hypothesis. This relationship is an interesting one, since it takes into account the different semantic models used by the languages, and also the differences in execution that arise from the translation.

6.4.1 Translating Configurations

We begin by introducing the notation that will be used in the following text: $\mathcal{C}_C$ and $\mathcal{C}_Q$ are the sets of CQP and QMC configurations, respectively; $\mathcal{P}_C$ and $\mathcal{P}_Q$ are the sets of CQP and QMC programs, respectively; the transitions $\longrightarrow_C$ and $\longrightarrow_Q$ denote the reduction relations on CQP and QMC configurations, respectively; $\longrightarrow^*$ is used to denote 0 or more transitions. For any partial mapping $g : A \rightharpoonup B$ and $a \in A, b \in B$ we define a partial mapping $g[a \mapsto b] : A \rightharpoonup B$ as $g[a \mapsto b](x) = b$ if $a = x$, and $g[a \mapsto b](x) = g(x)$ otherwise.

It is worth taking a moment at this point to describe the relationship between QMC programs and configurations, which is slightly more complicated than for CQP. A QMC configuration consists of a global store alongside a list of pairs containing the local store and process description. These processes are expressed using the *abstract syntax* of QMC, through which the operational semantics is defined. The translation

from concrete to abstract syntax is straightforward and we will not dwell on the distinction in the following.

The translation of a process assumes that we start from an empty state, however configurations, which describe processes during execution, also contain state information.

The extension from the syntactic translation, given by the function $\textsc{Tprog}[\![]\!]$, to the translation $\mathcal{T}$ is defined component-wise. Formally,

$$\mathcal{T}(([\widetilde{q} \mapsto |\psi\rangle]; \phi; P)) = eval_{VD}((\mathcal{T}_P(P), \mathcal{T}_\kappa(\widetilde{q}, \phi), \mathcal{T}_\Sigma(\widetilde{q}, P), |\psi\rangle))$$

where:

- $\mathcal{T}_P$ is the translation from CQP processes to QMC programs given by $\textsc{Tprog}[\![]\!]$. A QMC configuration contains the abstract program representation instead of the concrete representation resulting from $\textsc{Tprog}[\![]\!]$. This conversion is straightforward and is implicit in $\mathcal{T}_P$.

- $\mathcal{T}_\kappa$ populates the global store with channel and qubit identifiers using the CQP channel list ϕ and qubit list $\widetilde{q}$: $\mathcal{T}_\kappa(\widetilde{q}, \widetilde{c}) = o[q_1 \mapsto 1] \cdots [q_n \mapsto n][c_1 \mapsto \texttt{null}] \ldots [c_n \mapsto \texttt{null}]$ where $o : Id \rightharpoonup Val$ is an empty partial mapping from QMC identifiers to QMC values. The channel names occurring in ϕ and qubit names in $\widetilde{q}$ are those which have been declared in previous execution steps. Eventually, the global store must be populated with all variable names; this is achieved by $eval_{VD}$.

- $\mathcal{T}_\Sigma$ assigns an empty local store to each process. These stores will be populated by $eval_{VD}$ with any local variables.

- The purpose of $eval_{VD}$ is to populate the global and local stores with the remaining variables that cannot be determined directly from the CQP configuration. Through the analysis of the CQP process, $\textsc{Tprog}[\![]\!]$ produces a program with variable declaration statements. Beyond the first steps, these statements do not appear in executing programs. $eval_{VD}$ represents the evaluation of all variable declaration statements, and the result is two-fold; through this evaluation, the local and global stores are populated with all the variables that will appear in the program, and after this evaluation, the program will have the first proper statements at the top of the execution stack.

6.4.2 Execution Relationship

The overall relationship between configurations of CQP and QMC that we wish to prove is illustrated by the following diagram.

$$\begin{array}{ccc} C_C & \xrightarrow{*}_C & C'_C \\ \tau \downarrow & & \tau \downarrow \\ C_Q & \xrightarrow{*}_Q & C'_Q \end{array}$$

where $C_C, C'_C \in \mathcal{P}_C$ and $C_Q, C'_Q \in \mathcal{P}_Q$. This reflects the equivalence of an execution $(C_C \longrightarrow^*_C C'_C)$ of CQP and an execution $(C_Q \longrightarrow^*_Q C'_Q)$ of QMC. This relationship is based on executions, or sequences of transition. We now consider how single transitions fit into this picture, in order to use a co-inductive proof method.

In the ideal scenario, a single CQP transition would correspond to a single QMC transition. This is akin to strong simulation, and expands the above diagram to give the following:

$$\begin{array}{ccccccc} C_C & \xrightarrow{}_C & C^1_C & \xrightarrow{}_C & C^2_C & \longrightarrow & C^n_C \\ \tau \downarrow & & \tau \downarrow & & \tau \downarrow & \cdots & \tau \downarrow \\ C_Q & \xrightarrow{}_Q & C^1_Q & \xrightarrow{}_Q & C^2_Q & \longrightarrow & C^n_Q \end{array}$$

Unfortunately, the relationship between transitions is not as simple as this. There are two factors that we must consider; the different semantic models that are used, and the different execution models.

From Small–steps to Big–steps

The operational semantics of both languages are defined in terms of transitions from one configuration to another. In CQP the semantics are in the small–step style of Plotkin [2004], whilst the QMC semantics are in the big–step style of Kahn [1987].

In small-step semantics, a single execution step arises from the execution of an atomic element of the process. The steps involved in evaluating a complex expression will be a succession of individual steps evaluating one atomic element at a time. For example, the evaluation of addition may be defined in small–step semantics by the rules

$$\frac{e_1 \longrightarrow e'_1}{e_1 + e_2 \longrightarrow e'_1 + e_2} \qquad \frac{e_2 \longrightarrow e'_2}{v_1 + e_2 \longrightarrow v_1 + e'_2}$$

In this case, the evaluation of an expression $e_1 + e_2$ will result in several steps, depending on the form of e_1 and e_2,

$$e_1 + e_2 \longrightarrow v_1 + e_2 \longrightarrow v_1 + v_2 \longrightarrow v_3$$

where $v_3 = v_1 + v_2$.

Example 6.1. The evaluation of $2 + 3 + 1 + 6$ using small-step semantics.

$$\frac{2+3 \longrightarrow 5}{2+3+1+6 \longrightarrow 5+1+6} \qquad \frac{5+1 \longrightarrow 6}{5+1+6 \longrightarrow 6+6} \qquad 6+6 \longrightarrow 12$$

In contrast, big-step semantics give rise to the evaluation of a term in a single step. For example, using big–step semantics, we may find the rule

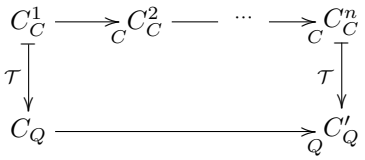
$$\frac{e_1 \longrightarrow v_1 \quad e_2 \longrightarrow v_2 \quad v_1 + v_2 \longrightarrow v_3}{e_1 + e_2 \longrightarrow v_3}$$

which leads to the evaluation of $e_1 + e_2$ in a single step, incorporating the individual evaluations of e_1 and e_2 as well as the addition.

Example 6.2. The evaluation of $2 + 3 + 1 + 6$ using big-step semantics.

$$\frac{\dfrac{2+3 \longrightarrow 5 \quad 5+1 \longrightarrow 6}{2+3+1 \longrightarrow 6} \qquad 6+6 \longrightarrow 12}{2+3+1+6 \longrightarrow 12}$$

Due to the different style of semantics in each language, we cannot expect each step of a CQP process execution to correspond to a single step in QMC. The difference between small-step and big-step operational semantics leads to a relationship in which several small-step transitions correspond to a single big-step transition. This is represented by the following diagram.



Semantic Equivalence

In addition to the different semantic models, we must also account for artifacts of the translation. For example, the assignment statements that are added for output actions and conditional unitary operators. The sequence of CQP executions does not correspond to QMC executions in these cases, because expressions are evaluated in a process context in CQP, but are evaluated sequentially in QMC.

If we consider a configuration $C = (\sigma; \phi; \{q \mathrel{*}= \mathsf{X}^{u+v}\}.P)$, then the translation begins with the assignment statement `X_cond:=u+v`, before the **`if`** construct. A transition from C, corresponding to the evaluation of the expression $u + v$, results in the configurations $C' = (\sigma; \phi; \{q \mathrel{*}= \mathsf{X}^{w}\}.P)$. The translation of this configura-

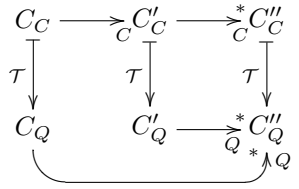


Figure 6.12. The requirement for a translation $\mathcal{T}$to be semantics-preserving.

tions begins with the statement `X_cond:=w`, again followed by the same **`if`** construct. Because QMC expressions are not executed in context, there is no execution step `X_cond:=u+v` $\longrightarrow$ `X_cond:=w`, that corresponds to the CQP step.

To account for such cases, we use a notion of *semantic equivalence*. By this, we mean that there exists a QMC configuration C_Q, such that $\mathcal{T}(C) \longrightarrow_Q^* C_Q$ and $\mathcal{T}(C') \longrightarrow_Q^* C_Q$.

To integrate this semantic equivalence with the conversion from small-step to big-step semantics, we also require C_Q to be the translation of a descendent CQP configuration. That is, $C_Q = \mathcal{T}(C'')$ where $C' \longrightarrow_C^* C''$. This results in the relationship expressed in Figure 6.12.

6.4.3 Preservation of Semantics

We now prove that the translation $\mathcal{T}$ preserves the semantics of CQP processes. By proving that the relationship illustrated in Figure 6.12 holds for all single step transitions $C_C \longrightarrow_C C'_C$, it follows that for any terminating execution $C_C \longrightarrow_C^* C''_C$ that $\mathcal{T}(C_C) \longrightarrow_Q^* \mathcal{T}(C''_C)$. This is precisely the requirement described at the beginning of this section.

We begin by considering the semantics of expressions in the following lemmas, before proving the main result (Theorem 6.4). These intermediate results require us to consider value and expression transitions in CQP, and the corresponding expression transitions in QMC. The CQP value and expression transitions ($\longrightarrow_v$ and $\longrightarrow_e$) have been described in Chapter 3 (Figure 3.3). These relations are defined on expression configurations; these are similar to configuration, but feature an expression in place of a process: $(\sigma; \phi; e)$. We denote a QMC expression transition using an uppercase E, the general form of which is $(e, \sigma, |\psi\rangle) \longrightarrow_E (v, |\psi'\rangle$.

Lemma 6.1. *Let e be defined by the grammar $e ::= \quad v \quad | \quad \mathsf{measure}\ q \quad | \quad e + e$. If $([\widetilde{q} \mapsto |\psi\rangle]; \phi; e) \longrightarrow_e ([\widetilde{q} \mapsto |\psi'\rangle]; \phi; v)$ then $(\textsc{Texpr}[\![e]\!], \sigma, |\psi\rangle) \longrightarrow_E (\textsc{Tval}[\![v]\!], |\psi'\rangle)$.*

Proof. R-Plus: We have the transition $(\sigma; \phi; u + v) \longrightarrow_e (\sigma; \phi; w)$ where $w = u + v$. The corresponding transition in QMC, since $\textsc{Tval}[\![u]\!] = u$ and$\textsc{Tval}[\![v]\!] = v$ is given by

$$(u + v, \sigma, |\psi\rangle) \longrightarrow_E (w, |\psi\rangle)$$

where $\textsc{Tval}[\![w]\!] = w$.

R-Measure: We have the transitions $([q, \widetilde{r} \mapsto |\psi\rangle]; \phi; \mathsf{measure}\ q) \longrightarrow_e \boxplus_i\ p_i \bullet ([q, \widetilde{r} \mapsto |\psi_i\rangle]; \phi; i) \overset{p_i}{\rightsquigarrow} ([q, \widetilde{r} \mapsto |\psi_i\rangle]; \phi; i)$, including the final probabilistic transition which results in the value i. The corresponding QMC transition for a measurement expression $\textsc{Texpr}[\![\mathsf{measure}\ q]\!]$ is

$$(\texttt{meas q}, \sigma, |\psi\rangle) \longrightarrow_E (i, |\psi_i\rangle)\ .$$

Because QMC only uses stabiliser states, the measurement outcomes are equi-probable and therefore it is not necessary to represent any explicit probabilities. □

Lemma 6.2. *Let e be defined by the grammar $e ::= v \mid \mathsf{measure}\ q \mid e + e$. If $([\widetilde{q} \mapsto |\psi\rangle]; \phi; e) \longrightarrow_e ([\widetilde{q} \mapsto |\psi'\rangle]; \phi; e')$ and $(\textsc{Texpr}[\![e']\!], \sigma, |\psi'\rangle) \longrightarrow_E (\textsc{Tval}[\![v]\!], |\psi''\rangle)$ then $(\textsc{Texpr}[\![e]\!], \sigma, |\psi\rangle) \longrightarrow_E (\textsc{Tval}[\![v]\!], |\psi''\rangle)$.*

Proof. We have the derivation

$$\frac{([\widetilde{q} \mapsto |\psi\rangle]; \phi; f) \longrightarrow_v ([\widetilde{q} \mapsto |\psi'\rangle]; \phi; v')}{([\widetilde{q} \mapsto |\psi\rangle]; \phi; E[f]) \longrightarrow_v ([\widetilde{q} \mapsto |\psi'\rangle]; \phi; E[v'])}$$

for some expression context E where $e = E[f]$ and $e' = E[v']$. By Lemma 6.1, we have $(\textsc{Texpr}[\![f]\!], \sigma, |\psi\rangle) \longrightarrow_E (\textsc{Tval}[\![v']\!], |\psi'\rangle)$. Therefore, we have the required derivation

$$\frac{(\textsc{Texpr}[\![f]\!], \sigma, |\psi\rangle) \longrightarrow_E (\textsc{Tval}[\![v']\!], |\psi'\rangle) \quad (\textsc{Texpr}[\![e']\!], \sigma, |\psi'\rangle) \longrightarrow_E (\textsc{Tval}[\![v]\!], |\psi''\rangle)}{(\textsc{Texpr}[\![e]\!], \sigma, |\psi\rangle) \longrightarrow_E (\textsc{Tval}[\![v]\!], |\psi''\rangle)}\ .$$

□

Lemma 6.3 (Preservation of Expression Semantics)**.** *Let e be defined by the grammar $e ::= v \mid \mathsf{measure}\ q \mid e + e$. If $([\widetilde{q} \mapsto |\psi\rangle]; \phi; e) \longrightarrow_e^* ([\widetilde{q} \mapsto |\psi'\rangle]; \phi; v)$ then $(\textsc{Texpr}[\![e]\!], \sigma, |\psi\rangle) \longrightarrow_E (\textsc{Tval}[\![v]\!], |\psi'\rangle)$.*

Proof. By induction on the length (n) of the sequence of transitions. The base case is $n = 0$, then $e = v$. For $n > 0$ we have

$$([\widetilde{q} \mapsto |\psi\rangle]; \phi; e) \longrightarrow_e ([\widetilde{q} \mapsto |\psi'\rangle]; \phi; e') \longrightarrow_e^* ([\widetilde{q} \mapsto |\psi''\rangle]; \phi; v)$$

where $\longrightarrow_e^*$ is a sequence of $n - 1$ transitions. The inductive hypothesis gives

$$(\textsc{Texpr}[\![e']\!], \sigma, |\psi'\rangle) \longrightarrow_E (\textsc{Tval}[\![v]\!], |\psi''\rangle)\ .$$

Then, by Lemma 6.2 we have the required QMC transition

$$(\textsc{Texpr}[\![e]\!], \sigma, |\psi\rangle) \longrightarrow_E (\textsc{Tval}[\![v]\!], |\psi''\rangle)\ .$$

□

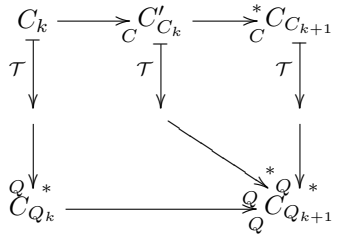
Theorem 6.4 (Preservation of Semantics)**.** *Let $C_C = (\sigma; \phi; P)$ and $C'_C = (\sigma'; \phi'; P')$. If $\Gamma \vdash P$ and $C_C \longrightarrow_C C'_C$ then there exists $C''_C = (\sigma''; \phi''; P'')$ such that $\mathcal{T}(C_C) \longrightarrow_Q^* \mathcal{T}(C''_C)$ and $\mathcal{T}(C'_C) \longrightarrow_Q^* \mathcal{T}(C''_C)$.*

Proof. By induction on the derivation of $C_C \longrightarrow_C C'_C$.

R-Expr: We have $P = F[e]$ and $P' = F[e']$ for a process context F, where $F \in \{v![[], \tilde{e}].P, \ldots, v![\tilde{v}, []], \{[]\}.P\}$. We exclude the contexts $[]?[\tilde{x}]$ and $[]![\tilde{e}]$ in which the hole must be filled with a channel name.
Subcase $F_k = c![v_1, \ldots, v_{k-1}, [], e_{k+1}, \ldots, e_n].P$: Let $C_{C_k} = ([\widetilde{q} \mapsto |\psi_k\rangle]; \phi; F_k[e_k])$ and $C'_{C_k} = \widetilde{q} = |\psi'_k\rangle; \phi; F_k[e'_k])$. We prove this by induction on the sequence of transitions $C_{C_1} \longrightarrow_C^* C'_{C_n}$. The inductive step considers a sequence of transitions $C_{C_k} \longrightarrow_C^* C_{C_{k+1}}$ where $F_{k+1}[e_{k+1}] = F_k[v_k]$, corresponding to the evaluation of the expression e_k that results in a value v_k. Let us define the QMC configuration

$$C_{Q_k} = ([c_k :\mapsto \textsc{Texpr}[\![e_k]\!]]; \ldots c_n := \textsc{Texpr}[\![e_n]\!]; c1!c_1; \ldots cn!c_n; c_{\text{ack}}?\text{ack}; \textsc{Tproc}[\![P]\!], \\ \kappa[c_1 \mapsto v_1, \ldots, c_{k-1} \mapsto v_{k-1}], \Sigma, |\psi_k\rangle) \ .$$

Diagrammatically, the relationship between the configurations that we are aiming to prove is as follows.



The base case is $k = n + 1$, then

$$\mathcal{T}(C_{C_{n+1}}) = (c_1 := v_1; \ldots c_n := v_n; c1!c_1; \ldots cn!c_n; c_{\text{ack}}?\text{ack}; \textsc{Tproc}[\![P]\!], \kappa, \Sigma, |\psi_1\rangle)$$

and $\mathcal{T}(C_{C_{n+1}}) \longrightarrow_Q^* C_{Q_{n+1}}$.

Assume for $0 < k \leq n$ that $C_{Q_{k+1}} \longrightarrow_Q C_{Q_{k+2}} \longrightarrow_Q \cdots C_{Q_{n+1}}$. Then we have

$$\mathcal{T}(C_{C_k}) = (c_1 := v_1; \ldots c_{k-1} := v_{k-1}; c_k := \textsc{Texpr}[\![e_k]\!]; \ldots c_n := \textsc{Texpr}[\![e_n]\!]; \\ c1!c_1; \ldots cn!c_n; c_{\text{ack}}?\text{ack}; \textsc{Tproc}[\![P]\!], \kappa, \Sigma, |\psi_k\rangle)$$

and

$$\mathcal{T}(C'_{C_k}) = (c_1 := v_1; \ldots c_{k-1} := v_{k-1}; c_k := \textsc{Texpr}[\![e'_k]\!]; \ldots c_n := \textsc{Texpr}[\![e_n]\!]; \\ c1!c_1; \ldots cn!c_n; c_{\text{ack}}?\text{ack}; \textsc{Tproc}[\![P]\!], \kappa, \Sigma, |\psi'_k\rangle) \ .$$

These QMC configurations give rise to the sequences of transitions $\mathcal{T}(C_{C_k}) \longrightarrow_Q^* C_Q$ where $C_Q = C_{Q_k}$ and $\mathcal{T}(C'_{C_k}) \longrightarrow_Q^* C'_Q$ where

$$C'_Q = (c_k := \textsc{Texpr}[\![e'_k]\!]; \ldots c_n := \textsc{Texpr}[\![e_n]\!]; c1!c_1; \ldots cn!c_n; c_{\text{ack}}?\text{ack}; \textsc{Tproc}[\![P]\!], \\ \kappa[c_1 \mapsto v_1, \ldots, c_{k-1} \mapsto v_{k-1}], \Sigma, |\psi'_k\rangle) \ .$$

Let $C''_{C_k} = ([\widetilde{q} \mapsto |\psi_{k+1}\rangle]; \phi; F_k[v_k]) = C_{C_{k+1}}$ where $C_{C'_k} \longrightarrow_C^* C''_{C_k}$. Then, by Lemma 6.3 we have transitions

$$(\textsc{Texpr}[\![e_k]\!], \kappa[c_1 \mapsto v_1, \ldots, c_{k-1} \mapsto v_{k-1}], |\psi_k\rangle) \longrightarrow_E (\textsc{Tval}[\![v_k]\!], |\psi_{k+1}\rangle) \qquad (6.1)$$

and

$$(\textsc{Texpr}[\![e'_k]\!], \kappa[c_1 \mapsto v_1, \ldots, c_{k-1} \mapsto v_{k-1}], |\psi'_k\rangle) \longrightarrow_E (\textsc{Tval}[\![v_k]\!], |\psi_{k+1}\rangle) \ . \qquad (6.2)$$

By (6.1) we can conclude $C_{Q_k} \longrightarrow_Q C_{Q_{k+1}}$ and also by (6.2) we have $C'_Q \longrightarrow_Q C_{Q_{k+1}}$. Subcase $F = \{[]\}.P$: We consider two cases; unitary operators and non-unitary expressions. Let $C_C = ([\widetilde{q} \mapsto |\psi\rangle]; \phi; F[e])$ where e is an expression defined by the grammar $e ::= v \mid \mathsf{measure}\ q \mid e + e$, then

$$\mathcal{T}(C_C) = (\textsc{Texpr}[\![e]\!]; \textsc{Tproc}[\![P]\!], \kappa, \sigma, |\psi\rangle) \ .$$

By Lemma 6.3 we have $C_C \longrightarrow_C^* C'_C$ where $C'_C = ([\widetilde{q} \mapsto |\psi'\rangle]; \phi; F[v])$ and $\mathcal{T}(C_C) \longrightarrow_Q^* (\textsc{Tval}[\![v]\!]; \textsc{Tproc}[\![P]\!], \kappa, \sigma, |\psi'\rangle) = \mathcal{T}(C'_C)$.

If e is a unitary operation, then we have $([\widetilde{q}, \widetilde{r} \mapsto |\psi\rangle]; \phi; \{\widetilde{q} *= \mathsf{U}^v\}.P) \longrightarrow_C^* ([\widetilde{qr} \mapsto |\psi'\rangle]; \phi; P)$. If $v = 0$ then $|\psi'\rangle = |\psi\rangle$ and if $v = 1$ then $|\psi'\rangle = (U \otimes I)|\psi\rangle$. We have

$\mathcal{T}(C_C) = (\texttt{U\_cond :=}\textsc{Tval}[\![v]\!]\texttt{; if :: (U\_cond = 0) -> break;}$
$\texttt{:: (U\_cond = 1) ->}\textsc{Tval}[\![U]\!]\textsc{Tval}[\![q_1]\!] \ldots \textsc{Tval}[\![q_n]\!]\texttt{; fi;}\ \textsc{Tproc}[\![P]\!], \kappa, \sigma, |\psi\rangle)$
$\longrightarrow_Q (\texttt{if :: (U\_cond = 0) -> break; :: (U\_cond = 1) ->}$
$\textsc{Tval}[\![U]\!]\textsc{Tval}[\![q_1]\!] \ldots \textsc{Tval}[\![q_n]\!]\texttt{; fi;}\ \textsc{Tproc}[\![P]\!], \kappa, \sigma[U\_cond \mapsto v], |\psi\rangle)$

If $v = 0$ then we have the transition

$$\longrightarrow_Q (\textsc{Tproc}[\![P]\!], \kappa, \sigma[U\_cond \mapsto v], |\psi\rangle)$$

otherwise, if $v = 1$ then we have the transition

$$\longrightarrow_Q (\textsc{Tproc}[\![P]\!], \kappa, \sigma[U\_cond \mapsto v], |\psi''\rangle)$$

This last transition is derived from

$$(\textsc{Tval}[\![\mathsf{U}]\!] q_1 \dots q_n, \kappa, \sigma, |\psi\rangle) \longrightarrow_E (\varepsilon, U_{qop}|\psi\rangle)$$

where U_{qop} is the unitary operator corresponding to the quantum operator $\textsc{Tval}[\![\mathsf{U}]\!]$. Because we are restricted to operators in the stabiliser formalism, then U_{qop} is defined and equal to $(U \otimes I)$. Therefore, $|\psi''\rangle = |\psi'\rangle$.

R-Com: Let $P = c![\widetilde{v}, \widetilde{q}].Q \parallel c?[\widetilde{x}{:}\widetilde{T}, \widetilde{y}{:}\widetilde{\mathsf{Qbit}}].R$ where $\widetilde{v}$ are non-qubit values and $\sigma = [q_1, \dots, q_r \mapsto |\psi\rangle]$. Then we have the transition

$$(\sigma; \phi; P) \longrightarrow_C (\sigma; \phi; Q \parallel R\{\widetilde{v}, \widetilde{q}/\widetilde{x}, \widetilde{y}\}) \ .$$

The translation of C_C is

$$\begin{aligned}\mathcal{T}(C_C) = (&c_1 := v_1; \dots, c_m := v_m; c_{m+1} := q_1; \dots, c_{m+n} := q_n; \\ &c1!c_1; \dots; c(m+n)!c_{m+n}; c_{\text{ack}}?\text{ack}; \textsc{Tproc}[\![Q]\!] \\ \parallel\ &c1?x_1; \dots cm?x_m; c(m+1)?y_1; \dots c(m+n)?y_n; c_{\text{ack}}!1; \textsc{Tproc}[\![R]\!], \\ &\kappa, (\sigma_Q, \sigma_R), |\psi\rangle)\end{aligned}$$

Let $C_Q = \mathcal{T}(C_C)$. The first transitions from this QMC configuration are the assignments to temporary variables $(c_1, \dots, c_{m+n})$ prior to sending.

$$\begin{aligned}C_Q \longrightarrow_Q\ &(c_2 := v_2; \dots, c_m := v_m; c_{m+1} := q_1; \dots, c_{m+n} := q_n; \\ &c1!c_1; \dots; c(m+n)!c_{m+n}; c_{\text{ack}}?\text{ack}; \textsc{Tproc}[\![Q]\!] \\ &\parallel c1?x_1; \dots cm?x_m; c(m+1)?y_1; \dots c(m+n)?y_n; c_{\text{ack}}!1; \textsc{Tproc}[\![R]\!], \\ &\kappa, (\sigma_Q[c_1 \mapsto v_1], \sigma_R), |\psi\rangle) \\ \longrightarrow_Q^*\ &(c_{m+1} := q_1; \dots, c_{m+n} := q_n; \\ &c1!c_1; \dots; c(m+n)!c_{m+n}; c_{\text{ack}}?\text{ack}; \textsc{Tproc}[\![Q]\!] \\ &\parallel c1?x_1; \dots cm?x_m; c(m+1)?y_1; \dots c(m+n)?y_n; c_{\text{ack}}!1; \textsc{Tproc}[\![R]\!], \\ &\kappa, (\sigma_Q[c_1 \mapsto v_1] \cdots [c_m \mapsto v_m], \sigma_R), |\psi\rangle)\end{aligned}$$

$$
\begin{aligned}
\longrightarrow_Q^* \;& (c1!c_1; \ldots; c(m+n)!c_{m+n}; c_{\text{ack}}?\text{ack}; \textsc{Tproc}[\![Q]\!] \\
& \parallel c1?x_1; \ldots cm?x_m; c(m+1)?y_1; \ldots c(m+n)?y_n; c_{\text{ack}}!1; \textsc{Tproc}[\![R]\!], \\
& \kappa, (\sigma_Q[c_1 \mapsto v_1] \cdots [c_m \mapsto v_m][q_1 \mapsto \mathsf{null}][c_{m+1} \mapsto 1] \\
& \quad \cdots [q_n \mapsto \mathsf{null}][c_{m+n} \mapsto n], \sigma_R), |\psi\rangle)
\end{aligned}
$$

The ordering of the next sequence of transitions, in which one process sends and the other receives the values, is non-deterministic due to the possible interleavings. We show one possible execution and note that all executions will arrive at the same final configuration.

$$
\begin{aligned}
\longrightarrow_Q \;& (c2!c_2; \ldots; c(m+n)!c_{m+n}; c_{\text{ack}}?\text{ack}; \textsc{Tproc}[\![Q]\!] \\
& \parallel c1?x_1; \ldots cm?x_m; c(m+1)?y_1; \ldots c(m+n)?y_n; c_{\text{ack}}!1; \textsc{Tproc}[\![R]\!], \\
& \kappa[c1 \mapsto v_1], (\sigma_Q[c_1 \mapsto v_1] \cdots [c_m \mapsto v_m][q_1 \mapsto \mathsf{null}][c_{m+1} \mapsto 1] \\
& \quad \cdots [q_n \mapsto \mathsf{null}][c_{m+n} \mapsto n], \sigma_R), |\psi\rangle) \\
\longrightarrow_Q \;& (c2!c_2; \ldots; c(m+n)!c_{m+n}; c_{\text{ack}}?\text{ack}; \textsc{Tproc}[\![Q]\!] \\
& \parallel c2?x_2; \ldots cm?x_m; c(m+1)?y_1; \ldots c(m+n)?y_n; c_{\text{ack}}!1; \textsc{Tproc}[\![R]\!], \\
& \kappa, (\sigma_Q[c_1 \mapsto v_1] \cdots [c_m \mapsto v_m][q_1 \mapsto \mathsf{null}][c_{m+1} \mapsto 1] \\
& \quad \cdots [q_n \mapsto \mathsf{null}][c_{m+n} \mapsto n], \sigma_R[x_1 \mapsto v_1]), |\psi\rangle) \\
\longrightarrow_Q^* \;& (c(m+1)!c_{m+1}; \ldots; c(m+n)!c_{m+n}; c_{\text{ack}}?\text{ack}; \textsc{Tproc}[\![Q]\!] \\
& \parallel c(m+1)?y_1; \ldots c(m+n)?y_n; c_{\text{ack}}!1; \textsc{Tproc}[\![R]\!], \\
& \kappa, (\sigma_Q[c_1 \mapsto v_1] \cdots [c_m \mapsto v_m][q_1 \mapsto \mathsf{null}][c_{m+1} \mapsto 1] \\
& \quad \cdots [q_n \mapsto \mathsf{null}][c_{m+n} \mapsto n], \sigma_R[x_1 \mapsto v_1] \cdots [x_m \mapsto v_m]), |\psi\rangle) \\
\longrightarrow_Q^* \;& (c_{\text{ack}}?\text{ack}; \textsc{Tproc}[\![Q]\!] \parallel c_{\text{ack}}!1; \textsc{Tproc}[\![R]\!], \\
& \kappa, (\sigma_Q[c_1 \mapsto v_1] \cdots [c_m \mapsto v_m][q_1 \mapsto \mathsf{null}] \cdots [q_n \mapsto \mathsf{null}], \\
& \sigma_R[x_1 \mapsto v_1] \cdots [x_m \mapsto v_m][y_1 \mapsto 1] \cdots [y_n \mapsto n]), |\psi\rangle)
\end{aligned}
$$

The final transitions are for the message acknowledgement, preventing Q from proceeding until the communication is complete.

$$
\begin{aligned}
\longrightarrow_Q^* \;& (\textsc{Tproc}[\![Q]\!] \parallel \textsc{Tproc}[\![R]\!], \\
& \kappa, (\sigma_Q[c_1 \mapsto v_1] \cdots [c_m \mapsto v_m][q_1 \mapsto \mathsf{null}] \cdots [q_n \mapsto \mathsf{null}][\text{ack} \mapsto 1], \\
& \sigma_R[x_1 \mapsto v_1] \cdots [x_m \mapsto v_m][y_1 \mapsto 1] \cdots [y_n \mapsto n]), |\psi\rangle)
\end{aligned}
$$

Finally, let us rename the variables in process R (we can take the substitution of

names inside $\textsc{Tproc}[\![R]\!]$) to give

$$\begin{aligned} C'_Q = \; & (\textsc{Tproc}[\![Q]\!] \parallel \textsc{Tproc}[\![R\{\widetilde{v}, \widetilde{q}/\widetilde{x}, \widetilde{y}\}]\!], \\ & \kappa, (\sigma_Q[c_1 \mapsto v_1] \cdots [c_m \mapsto v_m][q_1 \mapsto \mathsf{null}] \cdots [q_n \mapsto \mathsf{null}][\mathrm{ack} \mapsto 1], \\ & \sigma_R[v_1 \mapsto v_1] \cdots [v_m \mapsto v_m][q_1 \mapsto 1] \cdots [q_n \mapsto n]), |\psi\rangle) \end{aligned}$$

Because $\Gamma; \Sigma_Q, \widetilde{q} \vdash c![\widetilde{v}, \widetilde{q}].Q$, we have $\Gamma; \Sigma_Q \vdash Q$. Therefore $\mathcal{T}_\Sigma(Q, [q_1, \ldots, q_r \mapsto |\psi\rangle])$ does not include qubits $\widetilde{q}$, which are instead provided by $\mathcal{T}_\Sigma(R, [q_1, \ldots, q_r \mapsto |\psi\rangle])$. We have

$$\begin{aligned} \mathcal{T}(C'_C) = \; & (\textsc{Tproc}[\![Q]\!] \parallel \textsc{Tproc}[\![R\{\widetilde{v}, \widetilde{q}/\widetilde{x}, \widetilde{y}\}]\!], \kappa, \\ & (\sigma_Q[q_1 \mapsto \mathsf{null}] \cdots [q_n \mapsto \mathsf{null}], \sigma_R[q_1 \mapsto 1] \cdots [q_n \mapsto n]), |\psi\rangle) \end{aligned}$$

The temporary variables $c_1, \ldots, c_m$ do not appear in Q, hence we can conclude that $\mathcal{T}(C'_C) = C'_Q$.

R-Act: Let $P = \{v\}.Q$. Then $(\sigma; \phi; P) \longrightarrow_C (\sigma; \phi; Q)$. We have

$$\mathcal{T}(C_C) = (v; \textsc{Tproc}[\![Q]\!], \kappa; \Sigma; |\psi\rangle)$$

and the transition

$$\mathcal{T}(C_C) \longrightarrow_Q (\textsc{Tproc}[\![Q]\!], \kappa; \Sigma; |\psi\rangle) \; .$$

Therefore we have $\mathcal{T}(C_C) \longrightarrow_Q \mathcal{T}(C'_C)$.

R-Res: Let $P = (\nu c{:}\widehat{\ }[T])Q$. Then $(\sigma; \phi; P) \longrightarrow_C (\sigma; \phi, c; P)$ where we substitute the fresh channel name c for the variable c. Then we have the translations

$$\mathcal{T}(C_C) = (\textsc{Tproc}[\![Q]\!], \kappa, \Sigma, |\psi\rangle)$$

and

$$\mathcal{T}(C'_C) = (\textsc{Tproc}[\![Q]\!], \kappa, \Sigma, |\psi\rangle) \; .$$

In the first case, $c \in \kappa$ due to the use of an_G and $eval_{VD}$ which identifies the channel from the structure $(\nu c : \widehat{\ }[T])$ and then places it in the channel store through the evaluation of variable declarations. In the second case, $c \in \kappa$ comes directly from the CQP channel list using $\mathcal{T}_\kappa$.

R-Qbit: Let $P = (\mathsf{qbit}\ q)Q$. Then $([q_1, \ldots, q_n \mapsto |\psi\rangle]; \phi; P) \longrightarrow_C ([q_1, \ldots, q_n, q \mapsto |\psi\rangle|0\rangle]; \phi; Q)$. We have the translation

$$\mathcal{T}(C_C) = (q := \mathrm{newqubit}; \textsc{Tproc}[\![Q]\!], \kappa, \sigma, |\psi\rangle)$$

where σ contains the qubit mappings $q_i \mapsto i$ for $i \in I$ where $\Gamma; \{q_i | i \in I\} \vdash P$. Then

we have the transition

$$\mathcal{T}(C_C) \longrightarrow_Q (\textsc{Tproc}[\![Q]\!], \kappa, \sigma[q \mapsto n+1], |\psi\rangle|0\rangle)$$

which adds the new mapping $q \mapsto n+1$ to σ. We have $\Gamma; \{q_i | i \in I\}, q \vdash Q$, hence $\mathcal{T}(C_C) \longrightarrow_Q \mathcal{T}(C'_C)$.

R-Par: We have the derivation

$$\frac{(\sigma; \phi; Q) \longrightarrow_C (\sigma'; \phi'; Q')}{(\sigma; \phi; Q \parallel R) \longrightarrow_C (\sigma'; \phi'; Q' \parallel R)}.$$

The inductive hypothesis gives $C''_C = (\sigma''; \phi''; Q'')$ where $(\sigma; \phi; Q) \longrightarrow^*_C C''_C$ and $(\sigma'; \phi'; Q') \longrightarrow^*_C C''_C$ and $\mathcal{T}((\sigma; \phi; Q)) \longrightarrow^*_Q \mathcal{T}(C''_C)$ and $\mathcal{T}((\sigma'; \phi'; Q')) \longrightarrow^*_Q \mathcal{T}(C''_C)$. Then, by induction on the sequence of transitions, we get $\mathcal{T}((\sigma; \phi; Q \parallel R)) \longrightarrow^*_Q \mathcal{T}((\sigma''; \phi''; Q'' \parallel R))$ and $\mathcal{T}((\sigma'; \phi'; Q' \parallel R)) \longrightarrow^*_Q \mathcal{T}((\sigma''; \phi''; Q'' \parallel R))$. By applying R-Par on each transition, we get $(\sigma'; \phi'; Q' \parallel R) \longrightarrow^*_C (\sigma''; \phi''; Q'' \parallel R)$. □

6.5 Discussion

In this section, we consider the motivation and weaknesses of the translation presented in this chapter.

Our aim in defining this translation is to enable the verification of quantum protocols using process calculus and model checking. For these formal techniques to be effectively combined, it is vital that the respective specifications describe the same system. Proving that the translation preserves the semantics of CQP processes is intended to establish this equivalence.

We have described and proved the relationship that ensures a translated program can simulate all actions of the original CQP process. But does the translated program allow other behaviours that are not permitted in the CQP model? A particularly interesting case is the conversion from synchronous to asynchronous communication. In this case, constructs that would be evaluated by a single communication transition in CQP give rise to a series of assignment, output, and input actions. Combined with the addition of actions for acknowledgment, the number of transitions is over $3n$ times as long as the corresponding CQP actions for an n-part message. When concurrency is considered, there are many states that occur in the translated program which do not arise in the CQP process. Generally, the effect of this is to turn a communication that occurs at a particular time, into an action that occurs between two points in time.

More important than this "blurring" of time, is the order in which events occur. Events that must complete before the communication, are still required to do so, whilst

events that cannot occur until after are also maintained. There is one minor exception; in this translation, a sending process may begin output before a receiving process is ready. However, this isn't significant for two reasons. Firstly, because the sending process cannot continue until a receiver sends an acknowledgement, and secondly because this output doesn't constitute communication in the sense of synchronisation and transfer of information, which only occurs in the presence of a receiver.

The additional states that occur in a translated program are likely to affect the efficiency of verification. An increase in the size of the state space adds unwanted complexity, and uninformative extra behaviour. For this reason alone, it would be beneficial to avoid these states. A partial solution is offered by the prospect of "atomic" statements, which group a series of statements into a single execution step. Atomic statements appear in other modelling languages, such as Promela (the modelling language for the SPIN model checker [Holzmann 2003]). An atomic construct was a late addition to QMC and has not been included in the current translation. Syntactically, atomic constructs are represented by curly braces, hence we could translate $c![x, 3]$ to

```
{c_1 := x; c_2 := 3; c1!c_1; c2!c_2;} c_ack?ack;
```

Note that we cannot include `c_ack?ack` in the atomic construct because we cannot group the input and output actions of separate processes together. In effect, we are able to reduce the number of transitions from $3n$ times to 3 times the number of CQP transitions (excluding acknowledgment messages).

Implementation A software implementation of the translation has been implemented using the Java programming language, with the aim of integration with the QMC tool. The translator has been developed using the ANTLR parser generator [Parr 2007, 2008] and is currently available as a command line application.

6.6 Summary

Chapters 3 and 4 focussed on the analysis of quantum protocols using process calculus. In this chapter, we describe an approach that enables the combination of process calculus techniques with automated verification. This synthesis is achieved through a translation from CQP to the modelling language of the Quantum Model Checker software tool.

In Section 6.1, we describe the QMC tool that has been developed by Gay et al. [2008]. We briefly describe the stabilizer formalism, which is used by QMC to provide efficient simulation of quantum protocols. This is at the expense of universal quantum computation, which results in a significant restriction to the translation. The use of QCTL to express verification properties is described in Section 6.1.2.

The formal translation is defined in Section 6.2. The translation is defined inductively on the structure of CQP processes, following a similar approach to Nielson and Nielson [1999]. We demonstrate the translation in Section 6.3, by application to a random number generator and to the teleportation protocol.

In Section 6.4, we describe the requirements for a semantics preserving translation. This includes accounting for the difference between small-step and big-step semantics, as well as the use of auxiliary statements in the translation (such as the sequence of assignments prior to an output). We extend the syntactic translation of Section 6.2 to a translation of configurations, and then prove that this translation preserves the semantics of CQP processes.

7 Conclusion

This chapter begins with a brief summary of the achievements described in the previous chapters. This is followed in Section 7.2 by a conclusion of the work presented in this thesis. Finally, in Section 7.3 we outline several possible directions of study in which this work may be extended.

7.1 Summary

Chapter 1. The first chapter described the emergence of the quantum computation and communication discipline. We discussed the characteristics of quantum systems, highlighting the factors that differentiate this paradigm from classical information, such as entanglement and probabilistic measurement. We discussed the benefits of formal methods and presented a survey of recent work into the development of formal modelling and analysis techniques for quantum systems.

Chapter 2. A review of the background material necessary for the following chapters was provided, including the core concepts of quantum computation and an introduction to process calculus and bisimilarity.

Chapter 3. This chapter presented an investigation into observational equivalence for quantum processes, and addressed the issues regarding congruence for general quantum processes. The operational semantics of CQP was redefined using a labelled transition relation, in order to describe external interactions. We introduced a notion of behavioural equivalence for CQP processes, namely probabilistic branching bisimilarity. Using this relation, we showed that quantum teleportation is bisimilar to a quantum channel. This relation is not preserved by parallel composition in general, however we proved that for a particular class of processes including teleportation,

bisimilarity is preserved by parallel composition. We found that the implementation of measurement, based on probabilistic branching, leads to a description of the quantum state that disagrees with the theory of quantum mechanics. This provided the starting point for Chapter 4.

Chapter 4. In this chapter, we presented a new operational semantics for CQP which combines probabilistic branching with mixed quantum states, in order to address the incompatibility with quantum observations found in the previous chapter. We proved that the new transition relations preserve typing, and we redefined probabilistic branching bisimilarity with respect to the new semantics. We then proved that probabilistic branching bisimilarity is preserved by non-input, non-qubit contexts, and furthermore that full probabilistic branching bisimilarity is a congruence. Using this relation, we showed that quantum teleportation and superdense coding protocols are each congruent to their respective high-level specification processes.

Chapter 5. In this chapter, we presented an axiomatic approach to verification based on the full probabilistic branching bisimilarity from Chapter 4. Using teleportation as the motivation, we defined several equalities for the manipulation of quantum operators. Many of these rules are based on analogous principles from quantum mechanics. The principles of implicit and deferred measurement are two previously discussed examples, which have been presented as axioms in this chapter. We proved the soundness of the given axiomatisation, and we discussed the potential role of an expansion law and the difficulties in the adaptation of such a law.

Chapter 6. We defined a translation from CQP to the quantum model checking tool QMC. The purpose of this translation was to enable multiple formal techniques, both manual and automated, to be applied to a single specification. We investigated the relationship between the semantic models of the two languages, and we proved that the translation preserves the semantics of CQP processes, and is therefore suitable for formal reasoning.

7.2 Concluding Remarks

This thesis set out to further develop formal methods techniques for quantum information systems, and, in doing so, to improve our understanding of the principles of quantum mechanics in relation to communication and computation. A central part of this study focussed on the congruence properties of quantum systems, understanding why previous approaches have failed to produce congruence relations for general quantum systems, and improving these approaches to develop an interesting and practical notion of congruence.

The existing quantum process calculus Communicating Quantum Processes (CQP) has provided the foundation for much of this work, and the adaptation of CQP has successfully led to the achievement of these aims. Chapter 3 provides a deep understanding of the application of process calculus to quantum systems, in particular, focussing on the representation and manipulation of the quantum state. This understanding set out the foundations for Chapter 4, in which a radically new semantics of quantum process evolution was presented. An important factor in the development of this semantics was to ensure correspondence with quantum physical laws, most notably the interpretation of the density matrix representation of quantum state. This close tie with physical theory ensures that the calculus is able to model realistic systems.

Chapters 5 and 6 build upon the CQP framework by providing new methods for reasoning about quantum processes; such generalised results are the hallmark of formal methods. The equational theory presented in Chapter 5 is based upon the full probabilistic branching bisimilarity of Chapter 4 and illustrates the significance of congruence relations. The ability to prove equivalence using equational reasoning, instead of the longhand proofs seen in Chapters 3 and 4, significantly reduces the complexity of analysis and provides clear motivation for seeking a congruence relation. The translation described in Chapter 6 extends the facility of CQP in a separate direction, taking advantage of the Quantum Model Checker tool. The ability to combine the manual techniques of CQP with the automation offered by QMC through this translation offers a significant advantage for this analytical approach.

The application of formal methods to quantum systems is motivated in particular by their use in cryptographic applications. Secure communication networks based on quantum key distribution are the next step beyond point-to-point links, and their verification is likely to benefit from compositional as well as automated analysis where protocols such as quantum teleportation and superdense coding constitute the building blocks of large and complex systems. The equational theory of Chapter 5 and the translation of Chapter 6 both provide significant advances in this respect.

Feng et al. [2011] have recently presented an independent congruence relation for their process calculus qCCS. A detailed comparison between their congruence and the relation in Chapter 4 requires further study, however, as previously discussed, the semantics of qCCS processes differs from CQP in several respects. One advantages of this work is the ability to abstract from the number of auxiliary qubits required, enabling the 3-qubit teleportation protocol to be identified with a single-qubit channel. Additionally, qCCS does not have the extensible framework for the evaluation of expressions, which is likely to play a major role in the analysis of more complicated protocols. Conversely, the semantics of qCCS and the accompanying results are somewhat simpler, and thus will undoubtedly be very important in furthering

our understanding of the field. Nevertheless, while these advantages have come at the expense of additional complexity, the framework and results in this thesis retain the flexibility and extensibility of CQP, providing a solid foundation with which to progress.

7.3 Further Work

In this final section we outline several directions for future work based around CQP and the verification framework presented in this thesis.

Various extensions to the CQP language would facilitate the specification of the more complicated protocols. Examples given in [Gay and Nagarajan 2005, 2006] illustrate the potential use of "if-then-else" constructs, and structured data. We have also discussed the inclusion of recursion and replication, and the match operator. Some of these extensions may be achieved through encodings into the current calculus, while others require new primitives.

Also of interest would be an analysis of the recent work concerning the combination of probabilistic and non-deterministic transitions, and how this relates to process equivalence. Ideas from this field are likely to be relevant to the implementation of new process constructs in CQP, and may offer inspiration for improvements to the current implementation of probabilistic branching and non-deterministic choice in CQP.

The current software implementation of the translation would benefit from several improvements. These include the incorporation of type checking algorithms and robust variable analysis to detect type errors and alleviate the requirement for unique variable naming. The ability to convert nested processes to a named set would allow the choice between using named processes or the formal nested syntax, or even an arbitrary mixture. A significant benefit to the accessibility of the translation would be integration into the QMC tool; the current implementation has been developed in Java to facilitate this.

Papanikolaou [2009] outlines several improvements to QMC which would enhance the overall framework. Of particular significance, is the possibility to extend simulation to universal quantum computation. This is not possible to do in an efficient manner, however according to Aaronson and Gottesman [2004], stabilizer circuits can be extended to include a limited number of non-Clifford gates without completely losing the efficiency gain.

It would be interesting to integrate the results of Ying et al. [2007, 2009] on approximate bisimilarities with the concept of mixed configurations. Approximate bisimilarity is used to take into account quantum noise that may occur in physical implementations, and is therefore an important aspect for the formal verification

of quantum communication devices. The semantics of purely quantum processes in [Ying et al. 2007, 2009] uses mixed states, and the inclusion of probabilistic branching may provide the ability to incorporate classical information and obtain interesting congruence results.

There are many interesting directions for further study, and the hope is that the developments in this thesis will provide a solid grounding for future progress in quantum process calculus.

List of Abbreviations

CCS Calculus of Communicating Systems. 9

CQP Communicating Quantum Processes. 28

CTL Computation Tree Logic. 149

EQPL Exogenous Quantum Propositional Logic. 149

LTS labelled transition system. 24

qCCS Quantum CCS. 29

QCTL Quantum Computation Tree Logic. 148

QKD Quantum Key Distribution. 5

QMC Quantum Model Checker. 148

QPAlg Quantum Process Algebra. 28

QRAM quantum random access machine. 12

Index

Bibliography

S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004.

S. Abramsky and B. Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS)*. IEEE Computer Society, 2004. Also arXiv:quant-ph/0402130.

P. Adão and P. Mateus. A process algebra for reasoning about quantum security. In *Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005)*, volume 170 of *Electronic Notes in Theoretical Computer Science*, pages 3–21, March 2007. Preliminary version presented at QPL'05.

T. Altenkirch and J. Grattage. A functional quantum programming language. In *20th Annual IEEE Symposium on Logic in Computer Science*, pages 249–258, Jun 2005.

T. Altenkirch, J. Grattage, J. K. Vizzotto, and A. Sabry. An algebra of pure quantum programming. *Electronic Notes in Theoretical Computer Science*, 170:23 – 47, 2007. Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005).

S. Andova. Process algebra with probabilistic choice. In J.-P. Katoen, editor, *Formal Methods for Real-Time and Probabilistic Systems*, volume 1601 of *Lecture Notes in Computer Science*, pages 111–129. Springer Berlin / Heidelberg, 1999.

S. Andova and T. A. C. Willemse. Branching bisimulation for probabilistic systems: characteristics and decidability. *Theoretical Computer Science*, 356(3):325–355, 2006.

P. Baltazar, R. Chadha, and P. Mateus. Quantum Computation Tree Logic – model checking and complete calculus. *International Journal of Quantum Information*, 6 (2):219–236, 2008.

J. S. Bell. On the Einstein Podolsky Rosen paradox. *Physics*, 1(3):195–200, 1964.

C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179, Bangalore, India, December 1984.

C. H. Bennett, G. Brassard, and J.-M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin. Experimental quantum cryptography. *Journal of Cryptology*, 5:3–28, 1992a. 10.1007/BF00191318.

C. H. Bennett, G. Brassard, and N. D. Mermin. Quantum cryptography without Bell's theorem. *Physical Review Letters*, 68(5):557–559, Feb 1992b.

C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical Review Letters*, 70(13):1895–1899, Mar 1993.

C. H. Bennett, G. Brassard, C. Crpeau, and U. M. Maurer. Generalized privacy amplification. In *ISIT: Proceedings IEEE International Symposium on Information Theory, sponsored by The Information Theory Society of The Institute of Electrical and Electronic Engineers*, 1995.

J. A. Bergstra and J. W. Klop. Process algebra for synchronous communication. *Information and Control*, 60(1-3):109–137, 1984.

G. Berlín, G. Brassard, F. Brussières, and N. Godbout. Loss-Tolerant Quantum Coin Flipping. In *Second International Conference on Quantum, Nano and Micro Technologies (ICQNM 2008)*, pages 1–9, Feb 2008.

T. Bolognesi and E. Brinksma. Introduction to the ISO specification language LOTOS. *Computer Networks and ISDN Systems*, 14(1):25 – 59, 1987.

G. Brassard and C. Crépeau. Quantum bit commitment and coin tossing protocols. In A. Menezes and S. Vanstone, editors, *Advances in Cryptology-CRYPT0' 90*, volume 537 of *Lecture Notes in Computer Science*, pages 49–61. Springer Berlin / Heidelberg, 1991.

G. Brassard and L. Salvail. Secret-key reconciliation by public discussion. In *in Advances in Cryptology — EUROCRYPT '93*, pages 410–423. Springer-Verlag, 1994.

G. Brassard, S. L. Braunstein, and R. Cleve. Teleportation as a quantum computation. *Physica D: Nonlinear Phenomena*, 120:43–47, 1998.

E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 2000.

R. Cleaveland and S. Sims. Concurrency Workbench of the New Century (CWB-NC). `http://www.cs.sunysb.edu/~cwb/`, 2009.

V. Danos, E. D'Hondt, E. Kashefi, and P. Panangaden. Distributed measurement-based quantum computation. *Electronic Notes in Theoretical Computer Science*, 170:73 – 94, 2007a. Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005).

V. Danos, E. Kashefi, and P. Panangaden. The measurement calculus. *Journal of the ACM*, 54(2):8, 2007b.

T. Davidson, S. J. Gay, H. Mlnařík, R. Nagarajan, and N. Papanikolaou. Model checking for Communicating Quantum Processes. *International Journal of Unconventional Computing*, 8(1):73–98, 2012.

D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London Ser. A*, A400:97–117, 1985.

D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. In *Proc. R. Soc. Lond., Ser. A*, volume 439, pages 553–558, 1992.

E. D'Hondt. *Distributed quantum computation: a measurement-based approach*. PhD thesis, Vrije Universiteit Brussel, 2005.

D. Dieks. Communication by EPR devices. *Physics Letters A*, 92(6):271 – 272, 1982.

W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, Nov 1976.

P. A. M. Dirac. *The Principles of Quantum Mechanics*. Oxford University Press, fourth edition, 1958.

A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical Review*, 47(10):777–780, May 1935.

A. K. Ekert. Quantum cryptography based on Bell's theorem. *Physical Review Letters*, 67(6):661–663, Aug 1991.

M. Elboukhari, M. Azizi, and A. Azizi. Analysis of Quantum Cryptography Protocols by Model Checking. *International Journal of Universal Computer Sciences*, 1(1): 34–40, 2010.

T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *Information Theory, IEEE Transactions on*, 31(4):469–472, Jul 1985.

C. Elliott, A. Colvin, D. Pearson, O. Pikalo, J. Schlafer, and H. Yeh. Current status of the DARPA quantum network. arXiv:quant-ph/0503058v2, Mar 2005.

E. A. Emerson. *Temporal and modal logic*, volume B: Formal Models and Semantics, pages 995–1072. MIT Press, 1990.

Y. Feng, R. Duan, Z. Ji, and M. Ying. Probabilistic bisimilarities between quantum processes. arXiv:cs.LO/0601014, 2006.

Y. Feng, R. Duan, Z. Ji, and M. Ying. Probabilistic bisimulations for quantum processes. *Information and Computation*, 205(11):1608–1639, 2007.

Y. Feng, R. Duan, and M. Ying. Bisimulation for quantum processes. In *Proceedings of the 38th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '11, pages 523–534, New York, NY, USA, 2011. ACM.

R. Feynman. Simulating physics with computers. *International Journal of Theoretical*

Physics, 21(6&7):467–488, 1982.

W. Fokkink. *Introduction to Process Algebra*. Springer-Verlag, 2007.

S. Gay, R. Nagarajan, and N. Papanikolaou. Probabilistic model–checking of quantum protocols. arXiv:quant-ph/0504007, 2005.

S. J. Gay. Quantum programming languages: survey and bibliography. *Mathematical Structures in Computer Science*, 16(4):581–600, 2006.

S. J. Gay and R. Nagarajan. Communicating Quantum Processes. In *POPL '05: Proceedings of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 145–157, New York, NY, USA, 2005. ACM Press.

S. J. Gay and R. Nagarajan. Types and Typechecking for Communicating Quantum Processes. *Mathematical Structures in Computer Science*, 16(3):375–406, 2006.

S. J. Gay, N. Papanikolaou, and R. Nagarajan. QMC: a model checker for quantum systems. arXiv:0704.3705, 2007. Also Research Report RR432, Department of Computer Science, University of Warwick.

S. J. Gay, N. Papanikolaou, and R. Nagarajan. QMC: a model checker for quantum systems. In *CAV 2008: In Proceedings of the 20th International Conference on Computer Aided Verification*, volume LNCS of *Lecture Notes in Computer Science*, pages 543–547. Springer-Verlag, July 2008.

I. Glendinning. Links on simulation, modelling, and error prevention for quantum computers, 2010. `http://www.vcpc.univie.ac.at/~ian/hotlist/qc/simulation.shtml`.

L. K. Grover. A fast quantum mechanical algorithm for database search. In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, New York, NY, USA, 1996. ACM.

J. Gruska. *Quantum Computing*. McGraw Hill, 1999.

C. A. R. Hoare. Communicating Sequential Processes. *Communications of the ACM*, 21(8):666–677, 1978.

C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.

G. Holzmann. A Theory for Protocol Validation. *IEEE Transactions on Computers*, 100(31):730–738, 1982.

G. J. Holzmann. *The SPIN Model Checker: Primer and Reference Manual*. Addison-Wesley, 2003.

P. Jorrand and M. Lalire. Toward a quantum process algebra. In *CF '04: Proceedings of the 1st Conference on Computing Frontiers*, pages 111–119, New York, NY, USA, 2004. ACM Press.

G. Kahn. Natural semantics. In F. Brandenburg, G. Vidal-Naquet, and M. Wirsing,

editors, *STACS 87*, volume 247 of *Lecture Notes in Computer Science*, pages 22–39. Springer Berlin / Heidelberg, 1987. 10.1007/BFb0039592.

A. Kent. Quantum bit string commitment. *Physical Review Letters*, 90(23):237901, Jun 2003.

M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic symbolic model checker. In P. Kemper, editor, *Proc. Tools Session of Aachen 2001 International Multiconference on Measurement, Modelling and Evaluation of Computer-Communication Systems*, pages 7–12, September 2001. Available as Technical Report 760/2001, University of Dortmund.

M. Lalire. A probabilistic branching bisimulation for quantum processes. arXiv:quant-ph/0508116, 2005.

M. Lalire. Relations among quantum processes: bisimilarity and congruence. *Mathematical Structures in Computer Science*, 16(3):407–428, 2006.

M. Lalire and P. Jorrand. A process algebraic approach to concurrent and distributed quantum computation: Operational semantics. TUCS General Publication No 33, pages 109–126, Turku Centre for Computer Science, July 2004.

H.-K. Lo and H. F. Chau. Is quantum bit commitment really possible? *Physical Review Letters*, 78(17):3410–3413, Apr 1997.

G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 1055, pages 147–166. Springer-Verlag, Berlin Germany, 1996.

D. Markham and B. C. Sanders. Graph states for quantum secret sharing. *Physical Review A*, 78(042309):17, Oct 2008.

P. Mateus and A. Sernadas. Weakly complete axiomatization of Exogenous Quantum Propositional Logic. *Information and Computation*, 204(5):771—794, 2006.

D. Mayers. Unconditionally secure quantum bit commitment is impossible. *Physical Review Letters*, 78(17):3414–3417, Apr 1997.

D. Mayers. Unconditional security in quantum cryptography. *Journal of the ACM*, 48(3):351–406, 2001.

N. D. Mermin. From classical state-swapping to quantum teleportation. *Physical Review A*, 65(1):012320, Dec 2001.

R. Milner. *A Calculus of Communicating Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.

R. Milner. *Communication and Concurrency*. Prentice-Hall International Series in Computer Science. Prentice-Hall, Upper Saddle River, NJ, USA, 1989.

R. Milner. *Communicating and Mobile Systems: the Pi-Calculus*. Cambridge University Press, June 1999.

R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, I. *Information and Computation*, 100(1):1–40, 1992.

R. Nagarajan and S. J. Gay. Formal verification of quantum protocols. arXiv:quant-ph/0203086, 2002.

R. M. Needham and M. D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999, 1978.

M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

H. R. Nielson and F. Nielson. Semantics with applications: A formal introduction. Revised edition; original published by John Wiley & Sons, 1992, 1999.

N. Papanikolaou. Techniques for design and validation of quantum protocols. Master's thesis, Department of Computer Science, University of Warwick, 2004.

N. Papanikolaou. Definition of the QMC Specification Language. Available online at `http://www.dcs.warwick.ac.uk/~nikos/downloads/qmcsemantics.pdf`, October 2008.

N. Papanikolaou. *Model Checking Quantum Protocols*. PhD thesis, Department of Computer Science, University of Warwick, 2009.

D. Park. Concurrency and automata on infinite sequences. In *Proceedings of the 5th GI-Conference on Theoretical Computer Science*, pages 167–183, London, UK, 1981. Springer-Verlag.

T. Parr. *The Definitive ANTLR Reference: Building Domain-Specific Languages*. The Pragmatic Bookshelf, 2007.

T. Parr. ANTLR parser generator. `http://www.antlr.org`, Dec 2008.

S. Perdrix. Quantum patterns and types for entanglement and separability. In *Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005)*, volume 170 of *Electronic Notes in Theoretical Computer Science*, pages 125–138, 2007.

S. Perdrix. Quantum entanglement analysis based on abstract interpretation. In M. Alpuente and G. Vidal, editors, *Static Analysis*, volume 5079 of *Lecture Notes in Computer Science*, pages 270–282. Springer Berlin / Heidelberg, 2008.

G. D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI Report FN-19, Computer Science Department, Aarhus University, 1981.

G. D. Plotkin. The origins of structural operational semantics. In *Journal of Logic and Algebraic Programming*, pages 60–61, 2004.

A. Poppe, A. Fedrizzi, R. Ursin, H. Böhm, T. Lörunser, O. Maurhardt, M. Peev, M. Suda, C. Kurtsiefer, H. Weinfurter, T. Jennewein, and A. Zeilinger. Practical

quantum key distribution with polarization entangled photons. *Optics Express*, 12 (16):3865–3871, 2004.

F. Prost and C. Zerrari. A logical analysis of entanglement and separability in quantum higher-order functions. arXiv:0801.0649v1 [cs.LO], 2008.

M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, MIT Laboratory for Computer Science, Cambridge, MA 02139, Jan 1979.

E. Rieffel and W. Polak. An introduction to quantum computing for non-physicists. *ACM Computing Surveys*, 32(3):300–335, Sep 2000.

R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, February 1978.

A. W. Roscoe. *Model-checking CSP*, pages 353–378. Prentice Hall International (UK) Ltd., Hertfordshire, UK, UK, 1994.

R. Rüdiger. Quantum Programming Languages: An Introductory Overview. *The Computer Journal*, 50(2):134–150, 2007.

P. Ryan, S. Schneider, M. Goldsmith, G. Lowe, and B. Roscoe. *Modelling and Analysis of Security Protocols*. Addison Wesley, 2001.

D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.

P. Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14(04):527–586, 2004a.

P. Selinger. A brief survey of quantum programming languages. In Y. Kameyama and P. J. Stuckey, editors, *Functional and Logic Programming*, volume 2998 of *Lecture Notes in Computer Science*, pages 61–69. Springer Berlin / Heidelberg, 2004b.

P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *FOCS '94: Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Washington, DC, USA, 1994. IEEE Computer Society.

N. Trčka and S. Georgievska. Branching bisimulation congruence for probabilistic systems. *Electronic Notes in Theoretical Computer Science*, 220(3):129 – 143, 2008.

R. J. van Glabbeek and W. P. Weijland. Branching time and abstraction in bisimulation semantics. *Journal of the ACM*, 43(3):555–600, 1996.

S. Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, 1983.

W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299: 802—803, 1982.

A. K. Wright and M. Felleisen. A syntactic approach to type soundness. *Information and Computation*, 115(1):38–94, 1994.

M. Ying, Y. Feng, and R. Duan. An algebra of quantum processes. http://arxiv.org/abs/0707.0330v1, Jul 2007.

M. Ying, Y. Feng, R. Duan, and Z. Ji. An algebra of quantum processes. *ACM Transactions on Computational Logic*, 10(3):1–36, 2009.